Global Data Services

Developing Data-Intensive Applications Using Globus Software

Ian Foster

Computation Institute

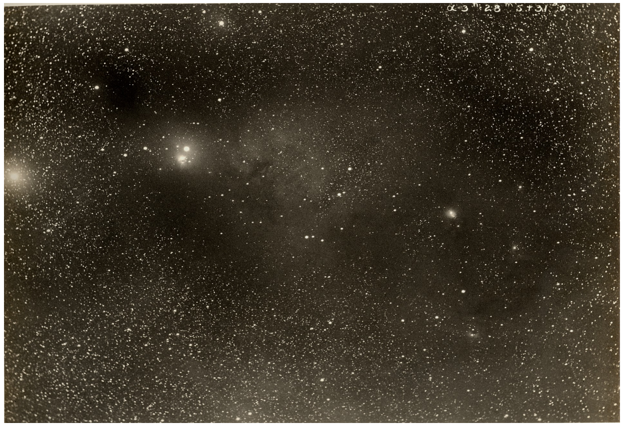
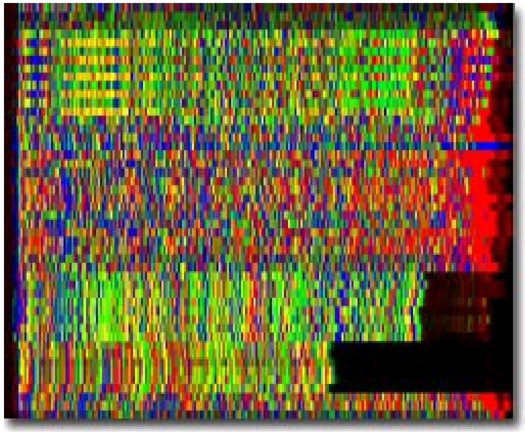
Argonne National Lab & University of Chicago

Acknowledgements

- Thanks to **Bill Allcock**, **Ann Chervenak**, **Neil P. Chue Hong**, **Mike Wilde**, and **Carl Kesselman** for slides
- I present the work of many Globus contributors: see **www.globus.org**
- Work supported by **NSF** and **DOE**

Context

- Science is increasingly about massive &/or complex data

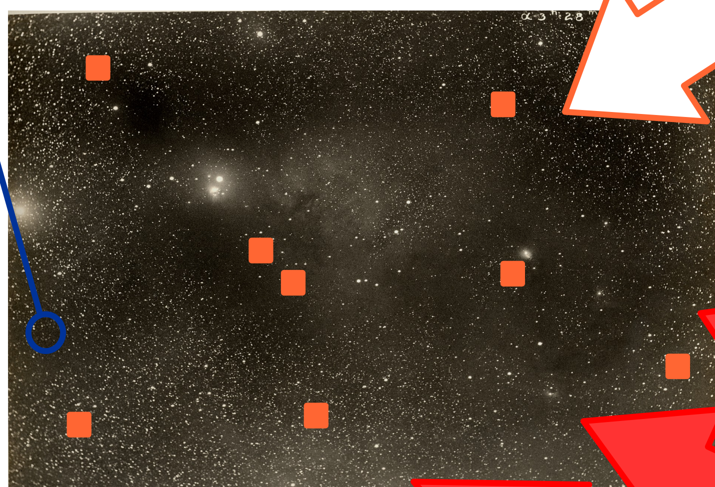




- Turning data into insight requires more than data access: we must connect data with people & computers

Science 1.0 → Science 2.0: For Example, Digital Astronomy

Tell me about this star

Tell me about these 20K stars



Support 1000s of users!!

E.g., Sloan Digital Sky Survey, ~40 TB; others much bigger soon

Data Challenges

- "Connecting data with people & computers"
  - Finding data of interest
  - Moving data to where it is needed
  - Managing large-scale computation
  - Scheduling resources on data
  - Managing who can access data when
- **Scaling** to address massive & distributed
  - Massive, distributed, & heterogeneous **data**
  - Massive & distributed **computation**
  - Massive & heterogeneous **workloads**
- Requires **global data services**

Global Data Services

- Deliver rich analysis capabilities on large & complex data—to distributed communities
  - Enable on-demand processing & analysis
  - Federate many (distributed) resources
  - Support (large) (distributed) communities
  - Manage ensemble to deliver performance
- Do so reliably and securely
- Scale to large data & computation
- Scale to large numbers of users

Overview

- Global data services
- Globus building blocks
- Building higher-level services
- Application case studies
- Summary

Overview

- Global data services
- **Globus building blocks**
  - **Overview**
  - **GridFTP**
  - **Reliable File Transfer Service**
  - **Replica Location Service**
  - **Data Access and Integration Services**
- Building higher-level services
- Application case studies
- Summary

Globus Software

- (Mostly Web Services) middleware providing key functionality relating to scaling
  - Access to data, and data movement
  - Authentication & authorization
  - Access to computation
  - Discovery and monitoring
- An enabler
  - Of solutions & tools for data access, distribution, and manipulation
  - Of infrastructures & applications

Grid Infrastructure: Open Standards/Software

Applications of the framework
(Compute, network, storage provisioning,
job reservation & submission, data management,
application service QoS, …)

Data Access & Management Services (DAIS, RFT, DRS)
Compute Access & Management Services (GRAM), etc.

WS-Resource Framework & WS-Notification*
(Resource identity, lifetime, inspection, subscription, …)

Web Services
(WSDL, SOAP, WS-Security, WS-ReliableMessaging, …)

*WS-Transfer, WS-Enumeration, WS-Eventing, WS-Management define similar functions

Available in High-Quality Open Source Software …

Globus Toolkit v4
www.globus.org

Security	Data Mgmt	Execution Mgmt	Info Services	Common Runtime
	Data Replication			
Credential Mgmt	Replica Location	Grid Telecontrol Protocol		
Delegation	Data Access & Integration	Community Scheduling Framework	WebMDS	Python Runtime
Community Authorization	Reliable File Transfer	Workspace Management	Trigger	C Runtime
Authentication Authorization	GridFTP	Grid Resource Allocation & Management	Index	Java Runtime

Building Blocks

- **Stage/move** large data to/from nodes
  - GridFTP, Reliable File Transfer (RFT)
  - Alone, and integrated with GRAM
- **Locate** data of interest
  - Replica Location Service (RLS)
- **Replicate** data for performance/reliability
  - Distributed Replication Service (DRS)
- Provide **access** to diverse data sources
  - File systems, parallel file systems, hierarchical storage: GridFTP
  - Databases: DAIS

Overview

- Global data services
- Globus building blocks
  - Overview
  - **GridFTP**
  - Reliable File Transfer Service
  - Replica Location Service
  - Data Access and Integration Services
- Building higher-level services
- Application case studies
- Summary

What is GridFTP?

- A secure, robust, fast, efficient, standards-based, widely accepted data transfer protocol
  - Independent implementations can interoperate
  - E.g., both the Condor Project and FermiLab have servers that work with ours
  - Many people have developed independent clients
- GT4 supplies a reference implementation:
  - Server
  - Client tools (globus-url-copy)
  - Development libraries

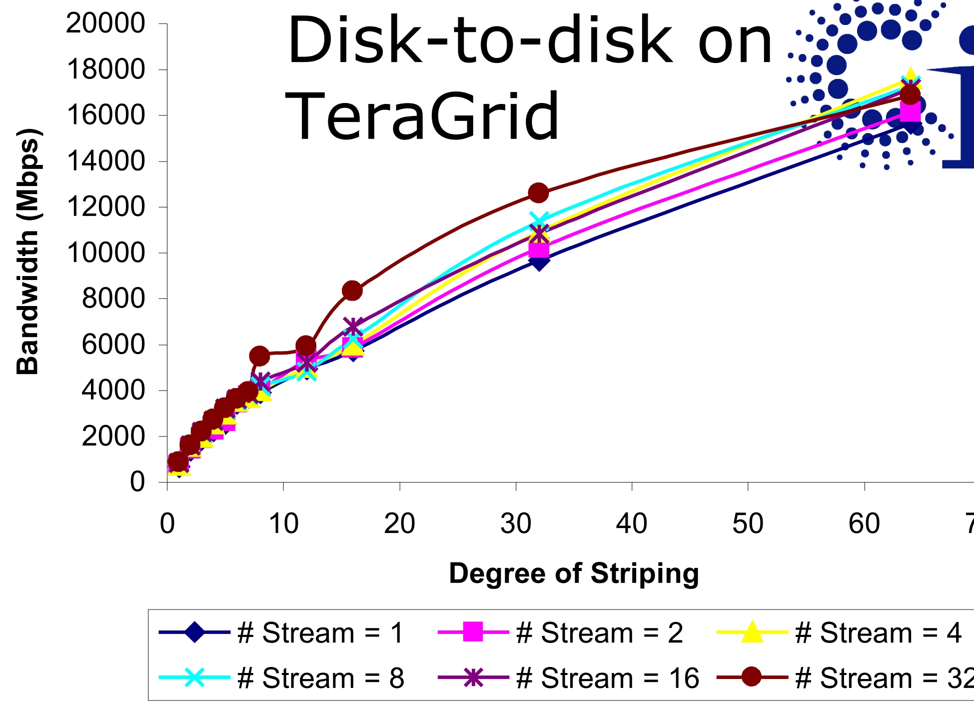
GridFTP: The Protocol

- FTP protocol is defined by several IETF RFCs
- Start with most commonly used subset
  - Standard FTP: get/put etc., 3rd-party transfer
- Implement standard but often unused features
  - GSS binding, extended directory listing, simple restart
- Extend in various ways, while preserving interoperability with existing servers
  - Striped/parallel data channels, partial file, automatic & manual TCP buffer setting, progress monitoring, extended restart

GridFTP: The Protocol (cont)

- Existing FTP standards
  - RFC 959: File Transfer Protocol
  - RFC 2228: FTP Security Extensions
  - RFC 2389: Feature Negotiation for the File Transfer Protocol
  - Draft: FTP Extensions
- New standard
  - GridFTP: Protocol Extensions to FTP for the Grid
  - Grid Forum Recommendation, GFD.20
  - www.ggf.org/documents/GWD-R/GFD-R.020.pdf

GridFTP in GT4



- 100% Globus code
  - No licensing issues
  - Stable, extensible
- IPv6 Support
- XIO for different transports
- Striping → multi-Gb/sec wide area transport
  - 27 Gbit/s on 30 Gbit/s link
- Pluggable
  - Front-end: e.g., future WS control channel
  - Back-end: e.g., HPSS, cluster file systems
  - Transfer: e.g., UDP, NetBLT transport

Striped Server Mode

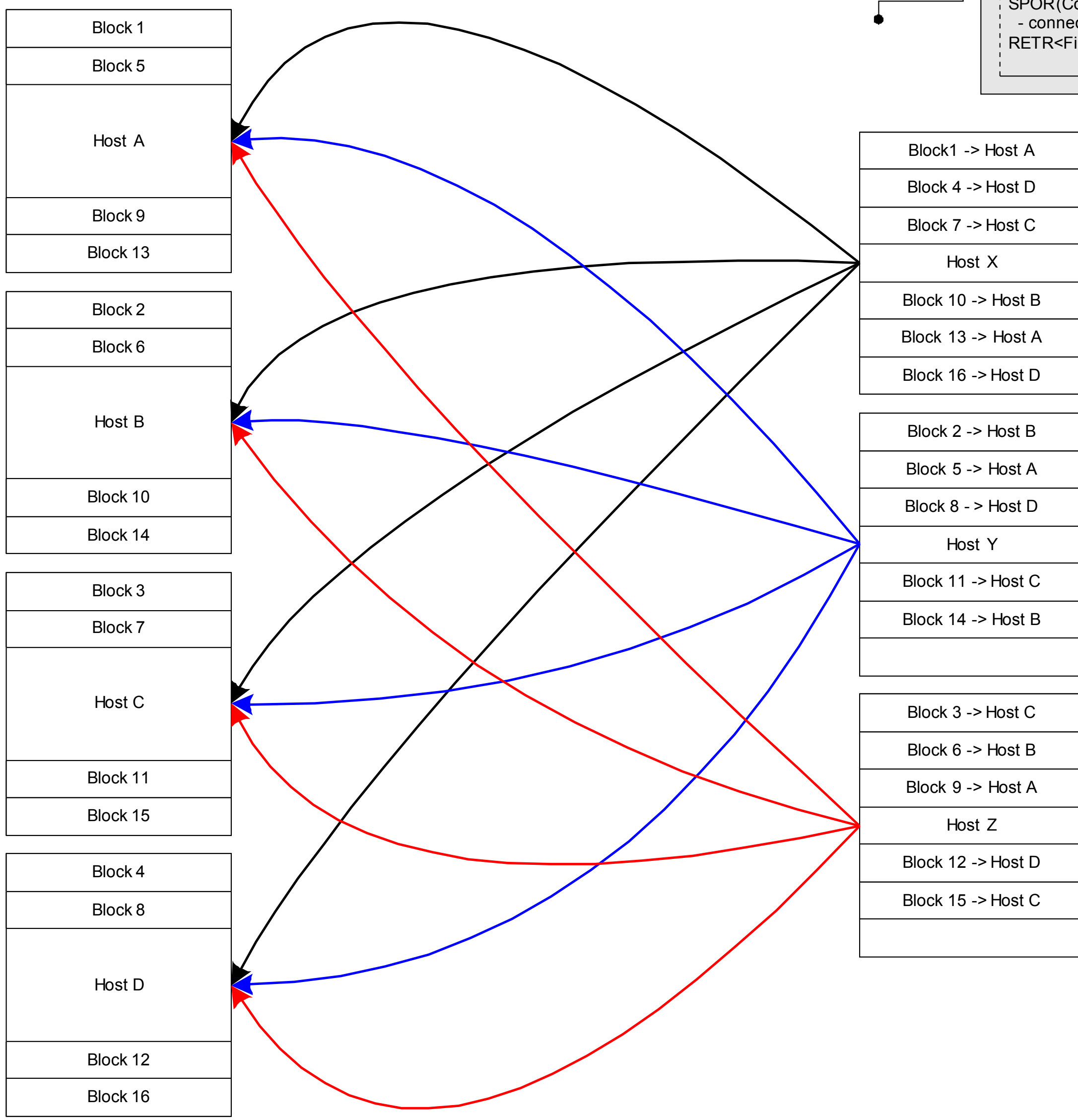
- Multiple nodes work together on a single file and act as a single GridFTP server
- Underlying parallel file system allows all nodes to see the same file system
  - Must deliver good performance (usually the limiting factor in transfer speed)—i.e., NFS does not cut it
- Each node then moves (reads or writes) only the pieces of the file for which it is responsible
- Allows multiple levels of parallelism, CPU, bus, NIC, disk, etc.
  - Critical to achieve >1 Gbs economically

18-Nov-03

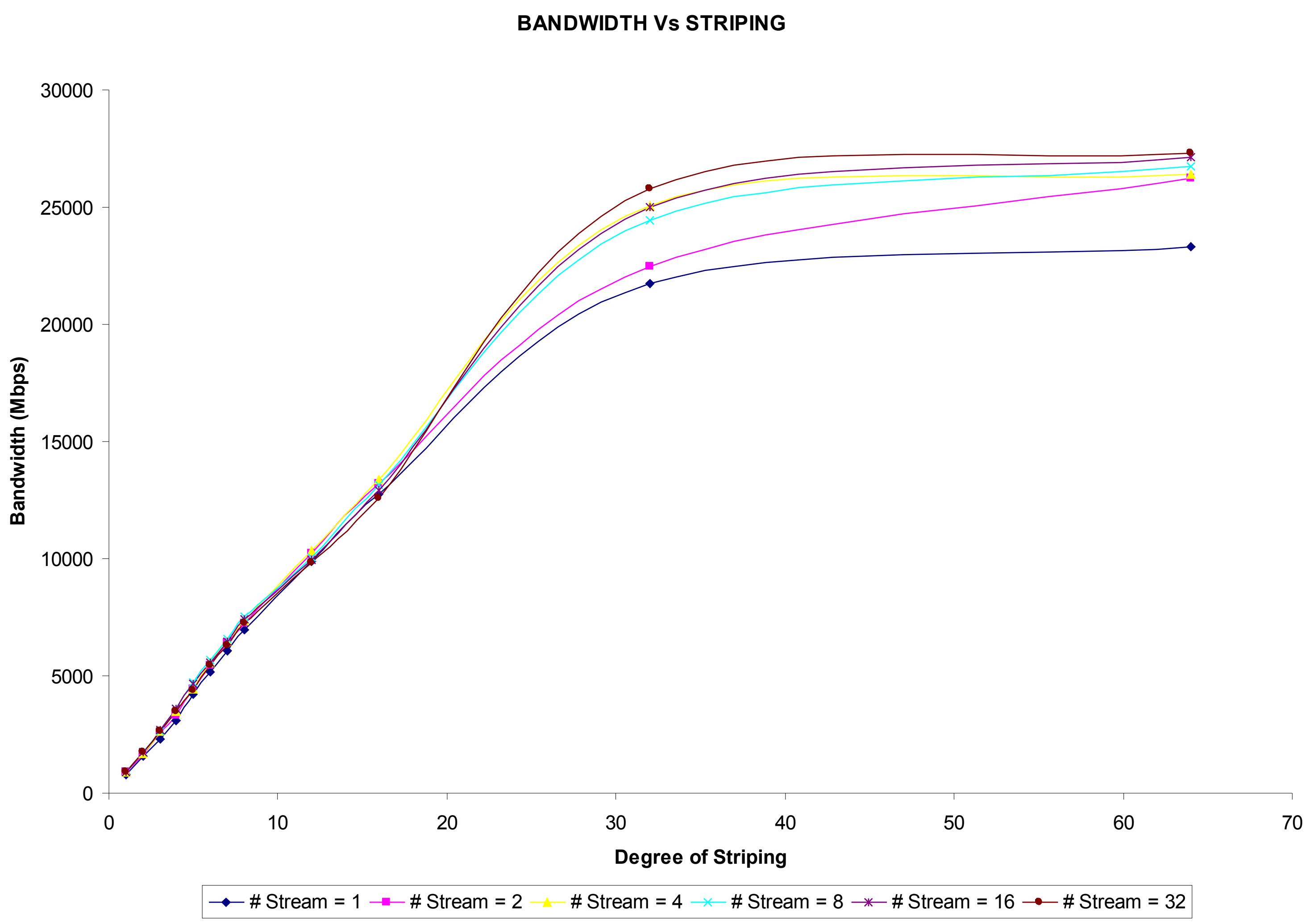
GridFTP Striped Transfer

MODE E
SPAS (Listen)
 - returns list of host:port pairs
STOR<FileName>

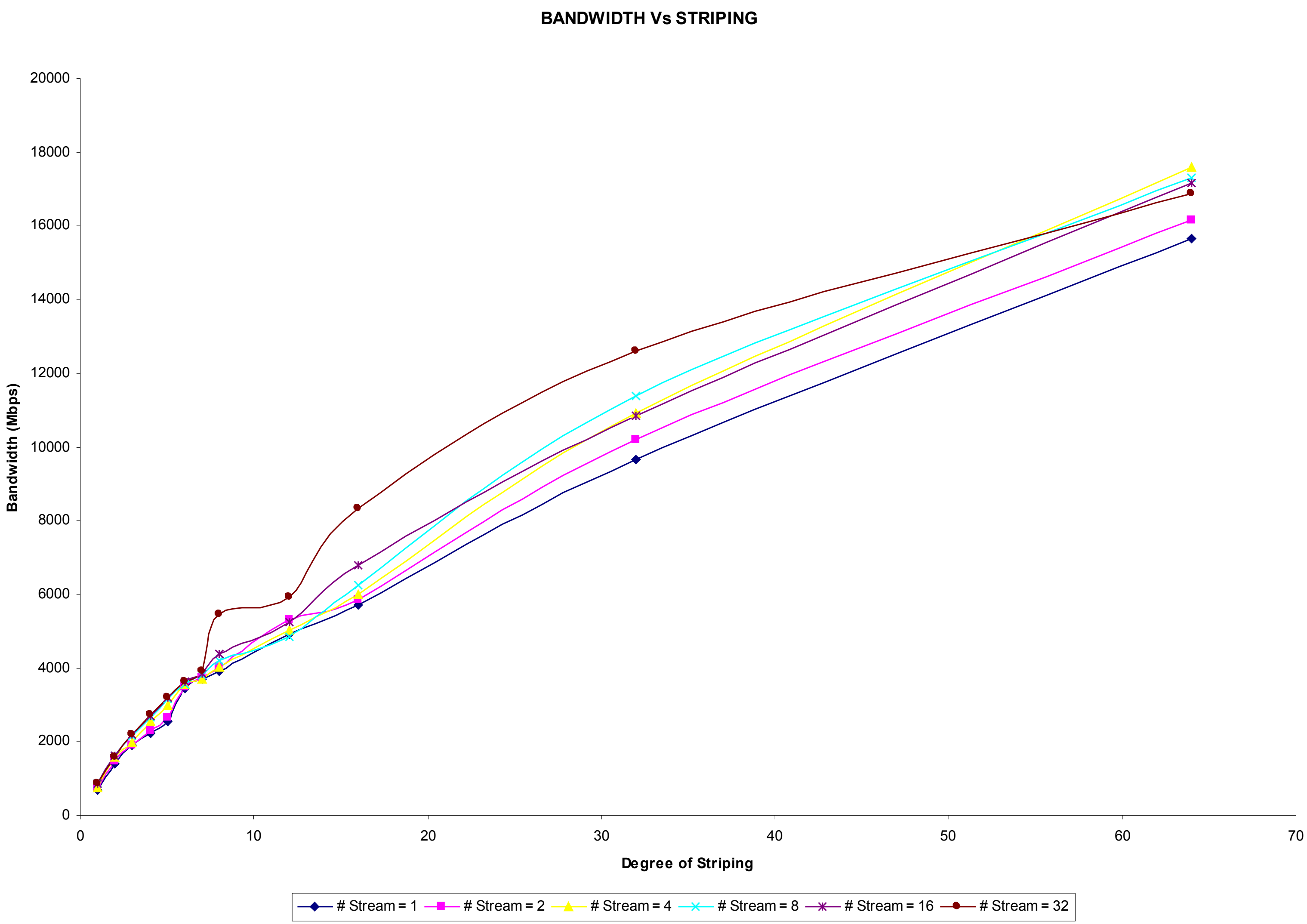
MODE E
SPOR(Connect)
 - connect to the host-port pairs
RETR<FileName>



Memory to Memory: TeraGrid



Disk to Disk: TeraGrid



New Server Architecture

- GridFTP (and normal FTP) use (at least) two separate socket connections:
  - A Control Channel for carrying the commands and responses
  - A Data Channel for actually moving the data
- Control Channel and Data Channel can be (optionally) completely separate processes.
- A single Control Channel can have multiple data channels behind it
- Future plans:
  - Load balancing proxy server
  - Dynamically created data movers

Data Transport Process Components

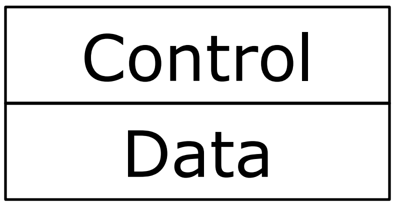
- The **protocol handler**. This part talks to the network and understands the data channel protocol
- **Data Storage Interface** (DSI). A well defined API that may be replaced to access things other than POSIX filesystems
- **ERET/ESTO processing**. Ability to manipulate the data prior to transmission.
  - Not implemented as a separate module for 4.0, but planned for 4.2

Data Storage Interfaces (DSIs)

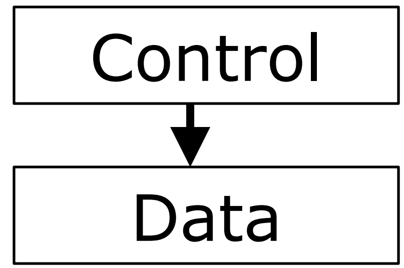
- Posix file I/O
- HPSS (with LANL / IBM)
- NeST (with UWis / Condor)
- SRB (with SDSC)

Possible Configurations

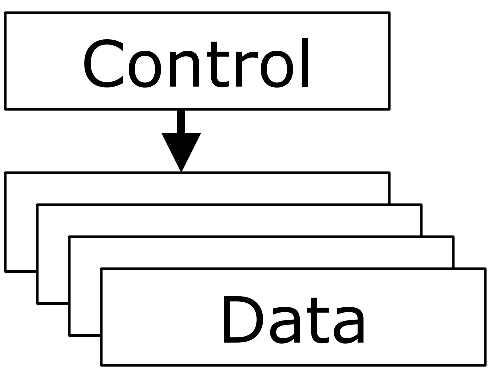
Typical Installation



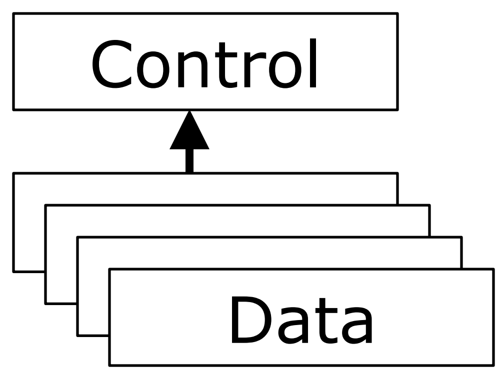
Separate Processes



Striped Server



Striped Server (future)



GridFTP: Caveats

- Protocol requires that the sending side do the TCP connect (possible Firewall issues)
  - Working on V2 of the protocol
    - Add explicit negotiation of streams to relax the directionality requirement above
    - Optionally adds block checksums and resends
    - Add a unique command ID to allow pipelining of commands
- Client / Server
  - Currently, no server library, therefore Peer to Peer applications are difficult
  - Generally needs a pre-installed server
    - Looking at a “dynamically installable” server

Extensible IO (XIO) system

- Provides a framework that implements a Read/Write/Open/Close Abstraction
- Drivers are written that implement the functionality (file, TCP, UDP, GSI, etc.)
- Different functionality is achieved by building protocol stacks
- GridFTP drivers allow 3rd party applications to access files stored under a GridFTP server
- Other drivers could be written to allow access to other data stores
- Changing drivers requires minimal change to the application code

Overview

- Global data services
- Globus building blocks
  - Overview
  - GridFTP
  - **Reliable File Transfer Service**
  - Replica Location Service
  - Data Access and Integration Services
- Building higher-level services
- Application case studies
- Summary

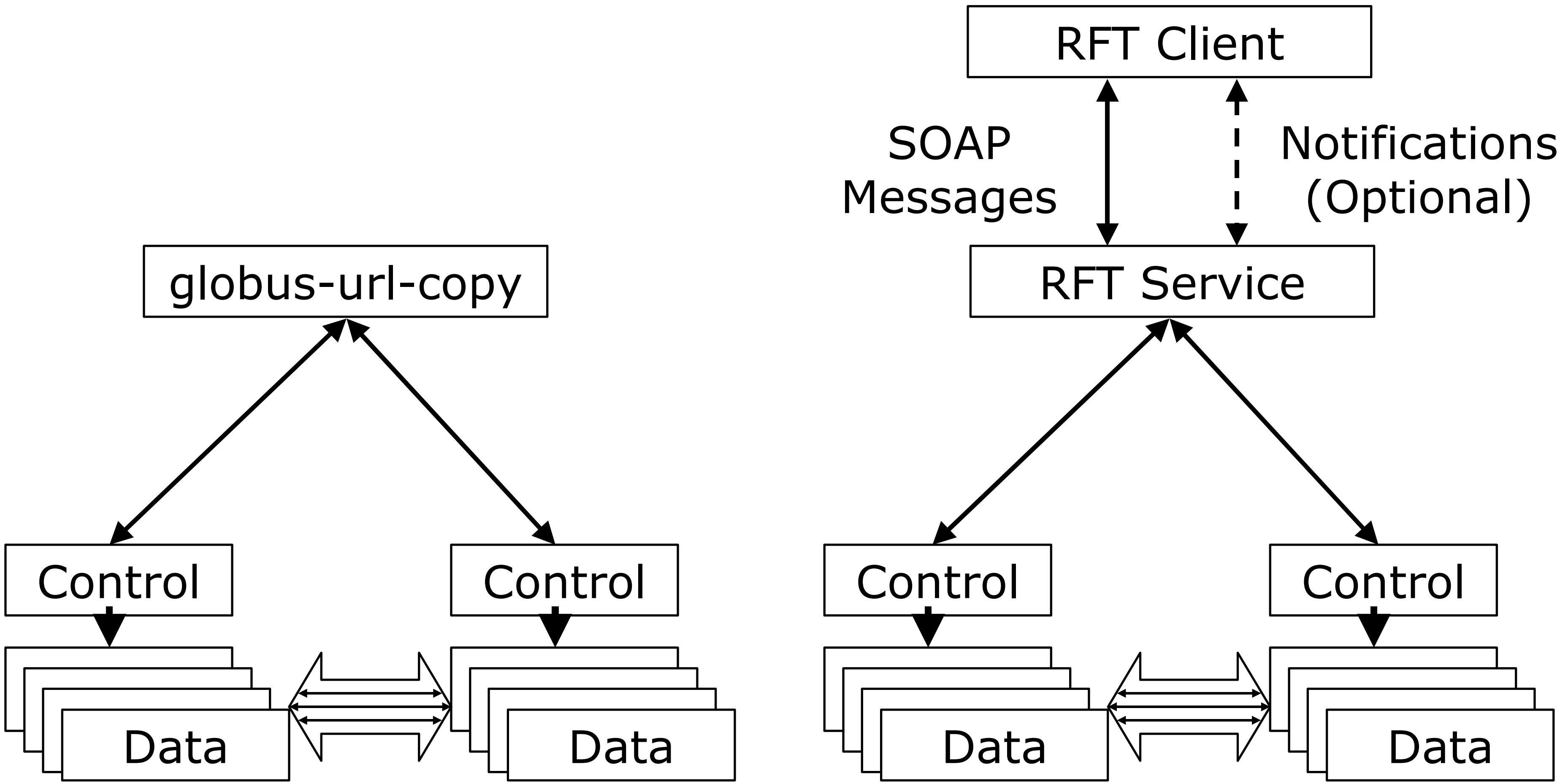
Reliable File Transfer

- Comparison with globus-url-copy
  - Supports all the same options (buffer size, etc)
  - Increased reliability because state is stored in a database.
  - Service interface: The client can submit the transfer request and then disconnect and go away
  - Think of this as a job scheduler for transfer job
- Two ways to check status
  - Subscribe for notifications
  - Poll for status (can check for missed notifications)

Reliable File Transfer

- RFT accepts a SOAP description of the desired transfer
- It writes this to a database
- It then uses the Java GridFTP client library to initiate 3rd part transfers on behalf of the requestor
- Restart Markers are stored in the database to allow for restart in the event of an RFT failure
- Supports concurrency, i.e., multiple files in transit at the same time, to give good performance on many small files

Data Transfer Comparison



the globus alliance
www.globus.org

Overview

- Global data services
- Globus building blocks
  - Overview
  - GridFTP
  - Reliable File Transfer Service
  - **Replica Location Service**
  - Data Access and Integration Services
- Building higher-level services
- Application case studies
- Summary

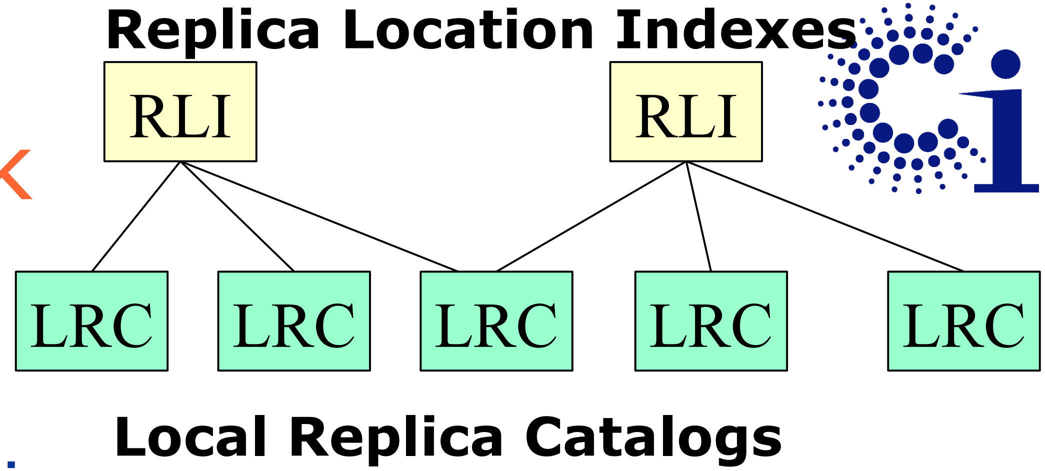
Replica Management

- Data intensive applications produce terabytes or petabytes of data
  - Hundreds of millions of data objects
- Replicate data at multiple locations for:
  - Fault tolerance: Avoid single points of failure
  - Performance: Avoid wide area data transfer latencies; load balancing

A Replica Location Service

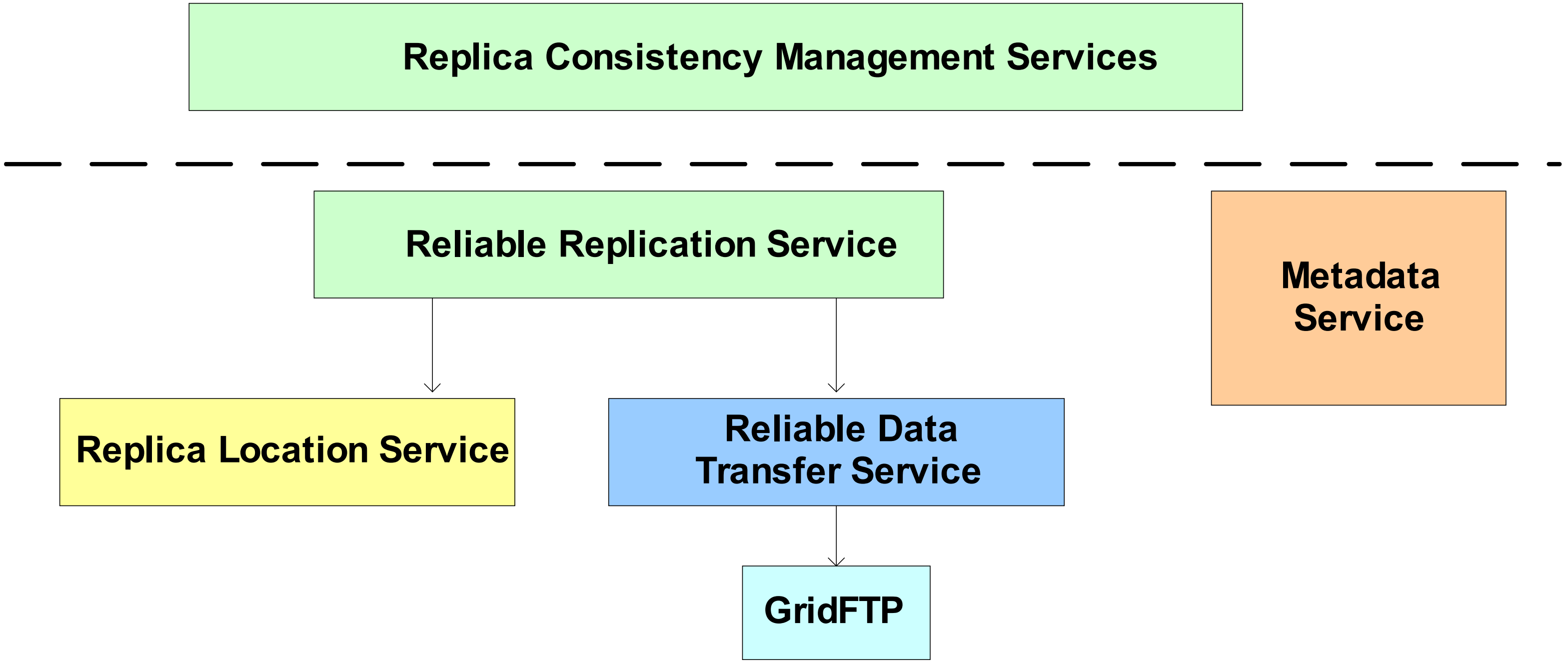
- A Replica Location Service (RLS) is a distributed registry that records the locations of data copies and allows replica discovery
  - RLS maintains mappings between logical identifiers and target names
  - Must perform and scale well: support hundreds of millions of objects, hundreds of clients
- RLS is one component of a Replica Management system
  - Other components include consistency services, replica selection services, reliable data transfer

RLS Framework



- Local Replica Catalogs (LRCs) maintain logical-to-target mappings
- Replica Location Index (RLI) node(s) aggregate information about LRC(s)
- LRCs use soft state updates to inform RLIs about their state: relaxed consistency
- Optional compression of state updates reduces communication, CPU, & storage costs
- Membership service registers participating LRCs and RLIs and deals with changes in membership

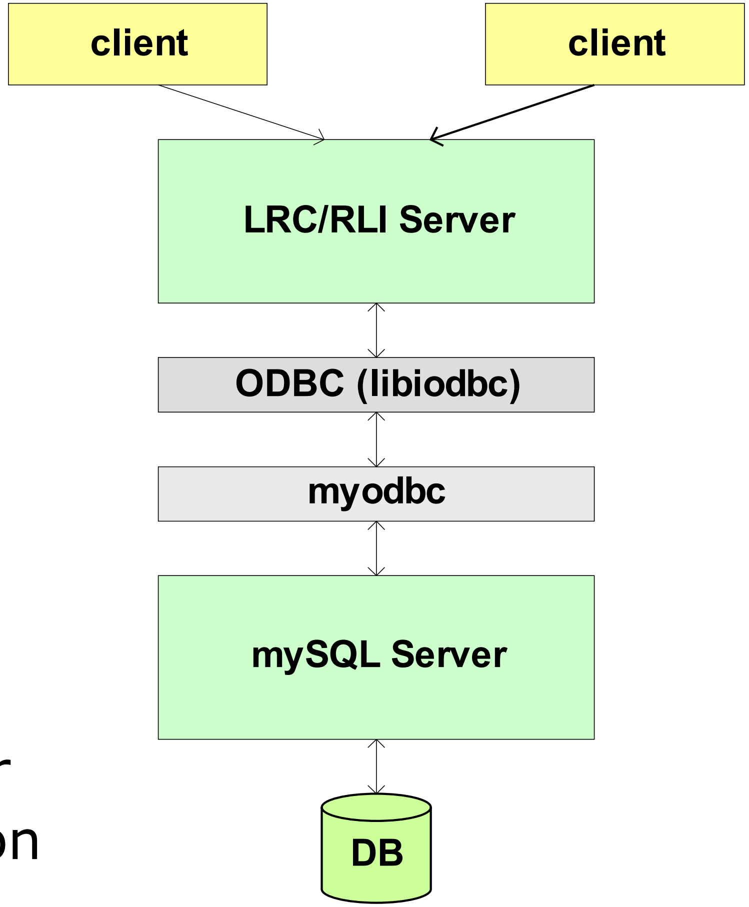
Replica Location Service In Context



- The Replica Location Service is one component in a layered data management architecture
- Provides a simple, distributed registry of mappings
- Consistency management provided by higher-level services

Components of RLS Implementation

- Common server implementation for LRC and RLI
- Front-End Server
  - Multi-threaded, written in C
  - GSI Authentication using X.509 certificates
- Back-end Server
  - MySQL or PostgreSQL Relational Database (later versions support Oracle)
  - No database back end required for RLIs using Bloom filter compression
- Client APIs: C and Java
- Client command-line tool



RLS Implementation Features

- Two types of soft state updates from LRCs to RLIs
  - Complete list of logical names registered in LRC
  - Compressed updates: Bloom filter summaries of LRC
- User-defined attributes
  - May be associated with logical or target names

RLS Implementation Features

- Soft state updates from LRCs to RLIs
  - Complete list of registered logical names
  - Compressed updates: Bloom filter summaries
- Immediate mode
  - Incremental updates
- User-defined attributes
  - May be associated with logical or target names
- Partitioning (without Bloom filters)
  - Divide soft state updates among RLI index nodes using pattern matching of logical names
- Currently, static membership configuration
  - No membership service

Soft State Update: (1) LFN List

- Send list of Logical Names stored on LRC
- Can do exact and wildcard searches on RLI
- Soft state updates get increasingly expensive as number of LRC entries increases
  - Space, network transfer time, CPU time on RLI
- E.g., with 1 million entries, takes 20 minutes to update mySQL on dual-processor 2 GHz machine (CPU-limited)

Soft State Update: (2) Bloom Filters

- Construct a summary of LRC state by hashing logical names, creating a bitmap
- Compression
- Updates much smaller, faster
- Supports higher query rate
- Small probability of false positives (lossy compression)
- Lose ability to do wildcard queries

the globus alliance
www.globus.org

Immediate Mode for Soft State Updates

- Immediate Mode
  - Send updates after 30 seconds (configurable) or after fixed number (100 default) of updates
  - Full updates are sent at a reduced rate
  - Tradeoff depends on volatility of data/frequency of updates
  - Immediate mode updates RLI quickly, reduces period of inconsistency between LRC and RLI content
- Immediate mode usually sends less data
  - Because of less frequent full updates

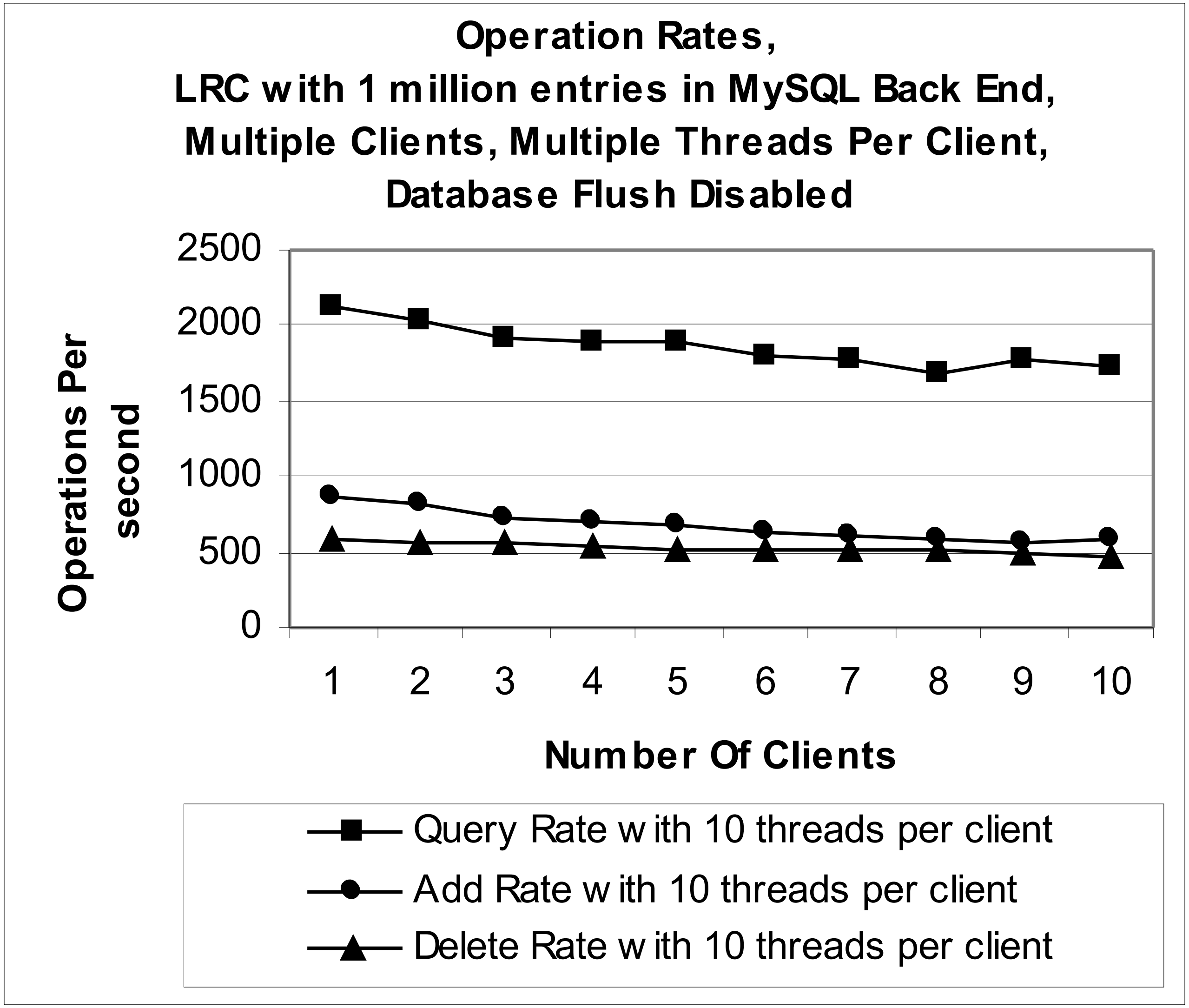
Performance Testing (see HPDC paper)

- Performance of individual LRC (catalog) or RLI (index) servers
  - Client program submits requests to server
- Performance of soft state updates
  - Client LRCs sends updates to index servers
- Software Versions:
  - Replica Location Service Version 2.0.9
  - Globus Packaging Toolkit Version 2.2.5
  - libiODBC library Version 3.0.5
  - MySQL database Version 4.0.14
  - MyODBC library (with MySQL) Version 3.51.06

Testing Environment

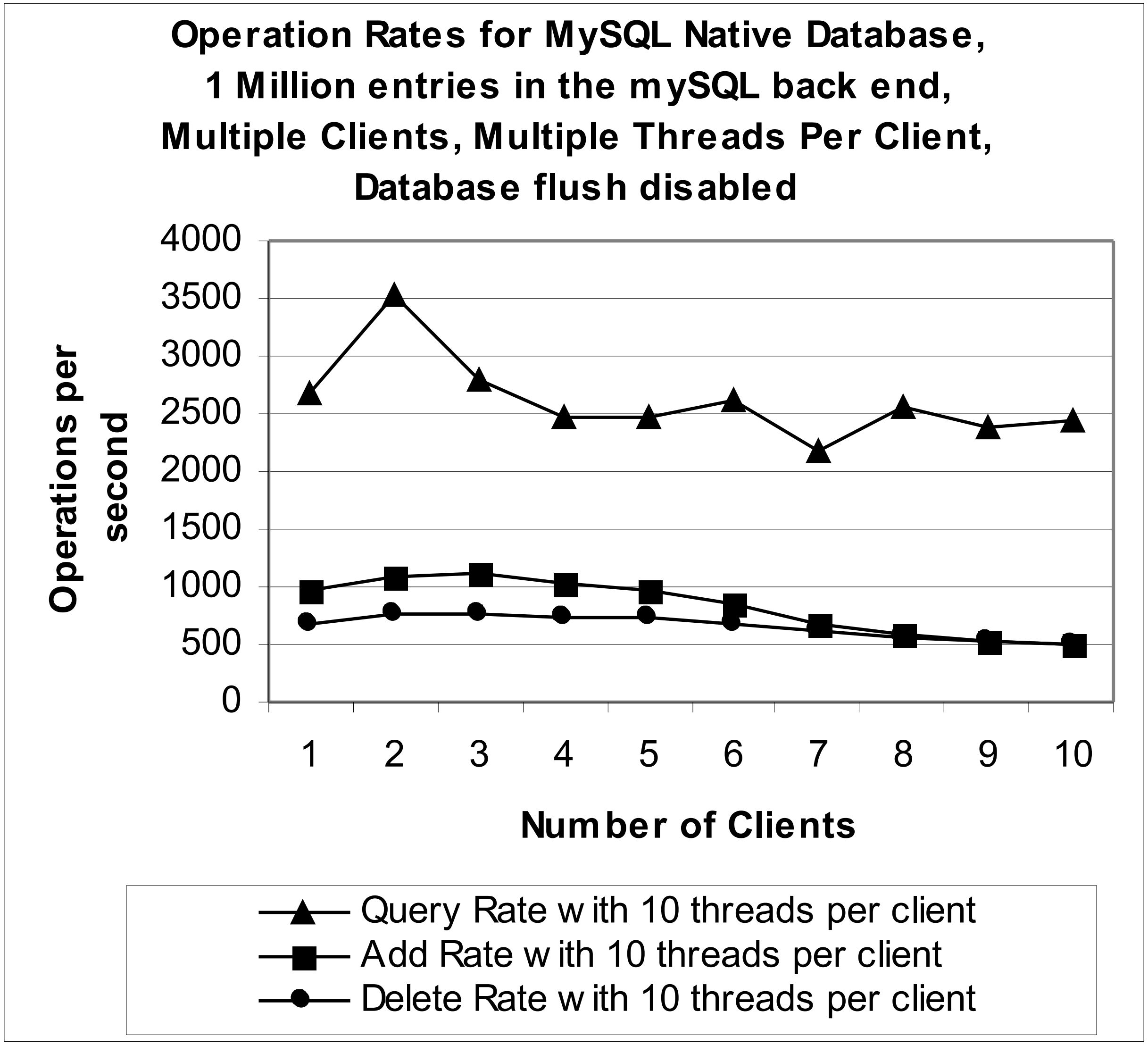
- Local Area Network Tests
  - 100 Megabit Ethernet
  - Clients (either client program or LRCs) on cluster: dual Pentium-III 547 MHz workstations with 1.5 GB memory running Red Hat Linux 9
  - Server: dual Intel Xeon 2.2 GHz processor with 1 GB memory running Red Hat Linux 7.3
- Wide Area Network Tests (Soft state updates)
  - LRC clients (Los Angeles): cluster nodes
  - RLI server (Chicago): dual Intel Xeon 2.2 GHz machine with 2 GB memory running Red Hat Linux 7.3

LRC Operation Rates (MySQL Backend)



- Up to 100 total requesting threads
- Clients and server on LAN
- Query: request the target of a logical name
- Add: register a new <logical name, target> mapping
- Delete a mapping

Comparison of LRC to Native MySQL Performance



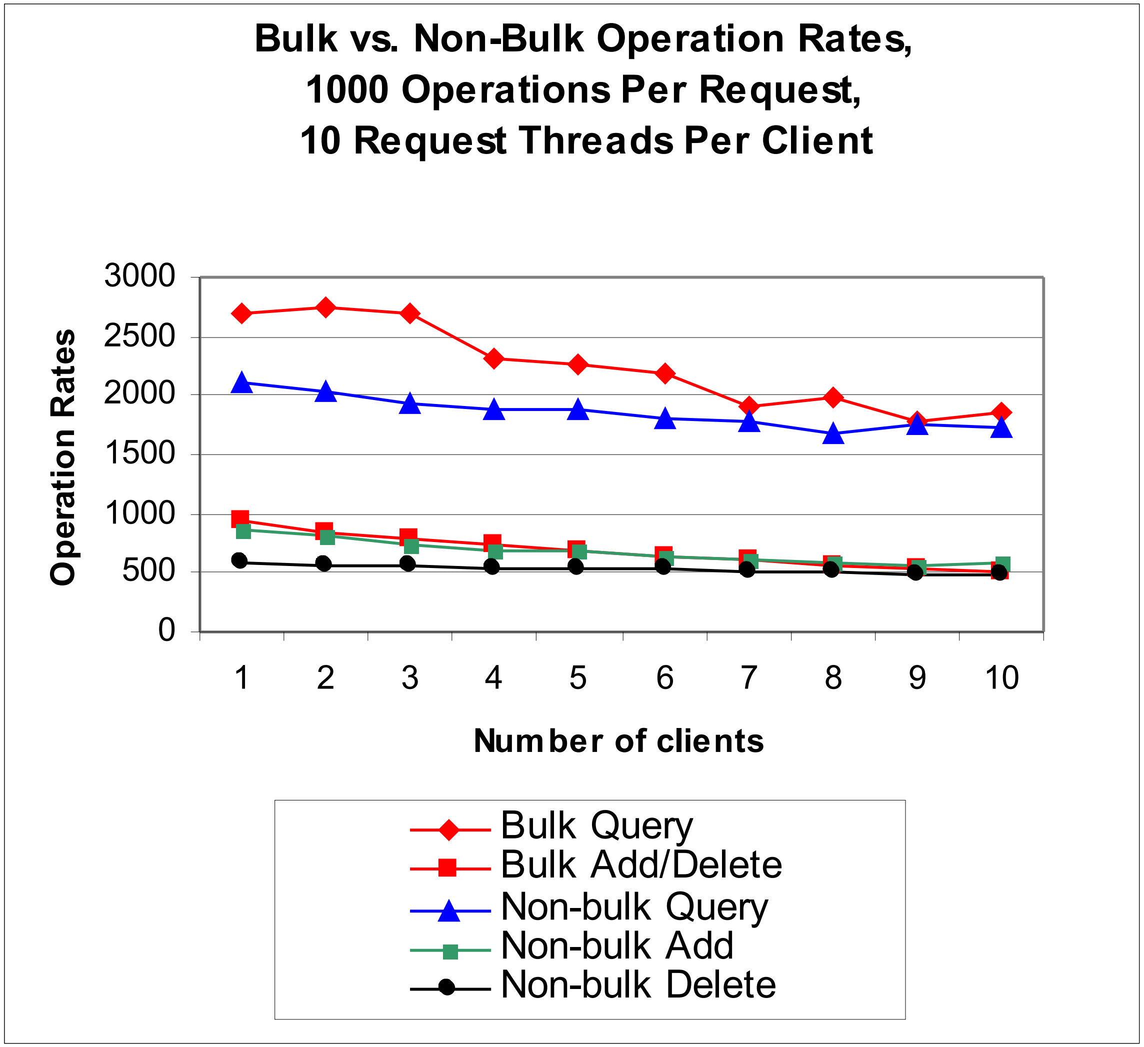
LRC Overheads

Highest for queries: LRC achieve 70-80% of native rates

Adds and deletes: ~90% of native performance for 1 client (10 threads)

Similar or better add and delete performance with 10 clients (100 threads)

Bulk Operation Performance



- For user convenience, server supports bulk operations
- E.g., 1000 operations per request
- Combine adds/deletes to maintain approx. constant DB size
- For small number of clients, bulk operations increase rates
- E.g., 1 client (10 threads) performs 27% more queries, 7% more adds/deletes

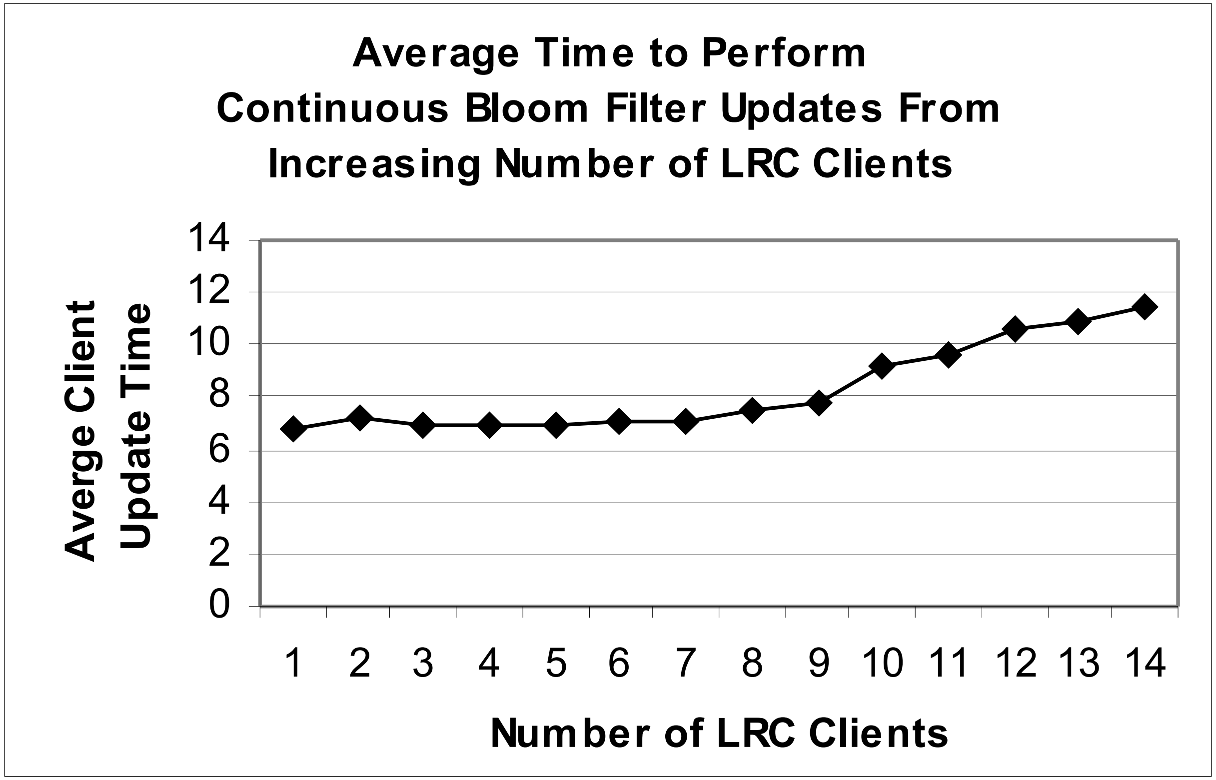
Bloom Filter Compression

- Construct a summary of each LRC's state by hashing logical names, creating a bitmap
- RLI stores in memory one bitmap per LRC
- Advantages:
  - Updates much smaller, faster
  - Supports higher query rate (satisfied from memory rather than database)
- Disadvantages:
  - Lose ability to do wildcard queries, since not sending logical names to RLI
  - Small probability of false positives (configurable)
  - Relaxed consistency model

Bloom Filter Performance: Single Wide Area Soft State Update (Los Angeles to Chicago)

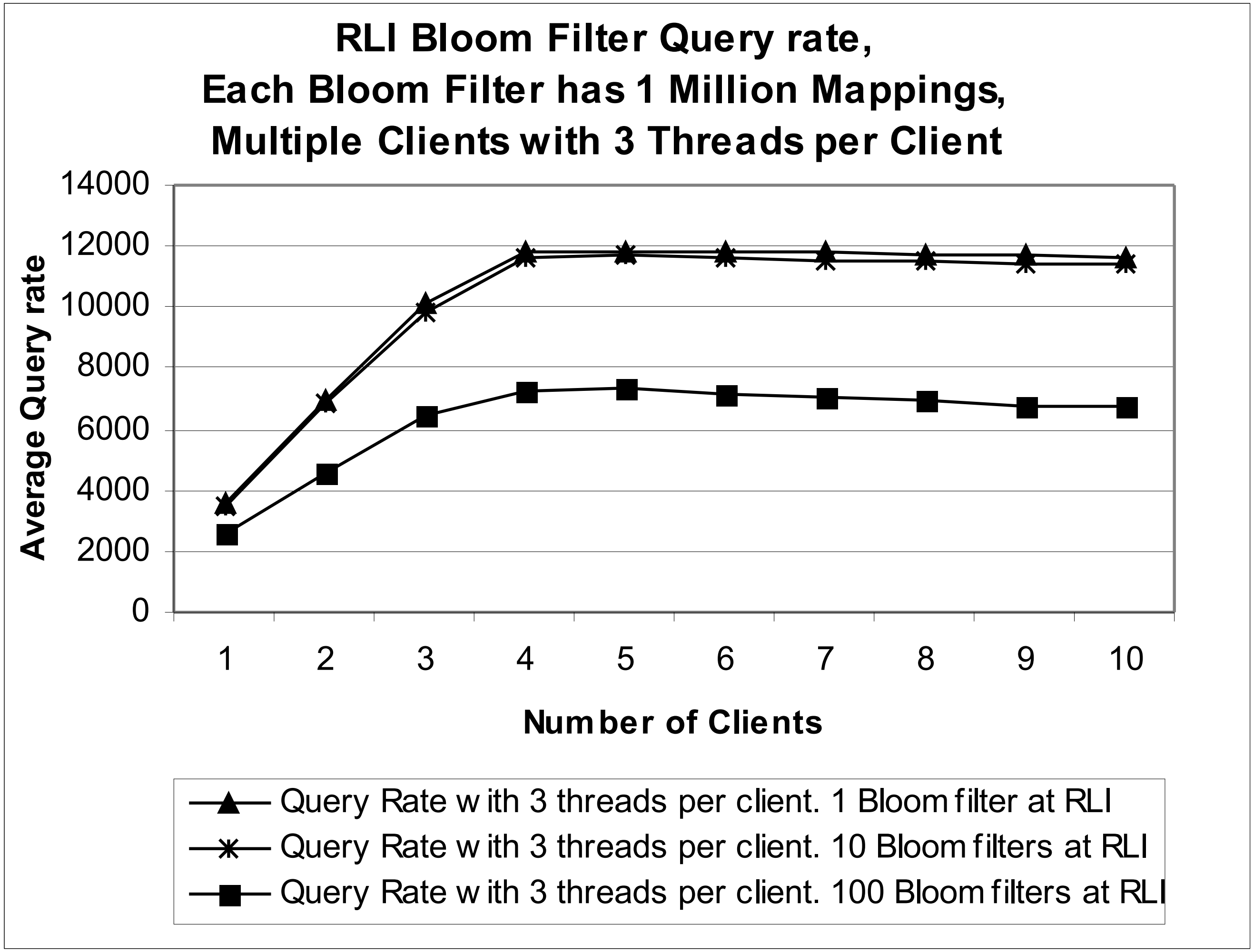
LRC Database Size	Avg. time to send soft state update (seconds)	Avg. time for initial bloom filter computation (seconds)	Size of bloom filter (bits)
100,000 entries	Less than 1	2	1 million
1 million entries	1.67	18.4	10 million
5 million entries	6.8	91.6	50 million

Scalability of Bloom Filter Updates



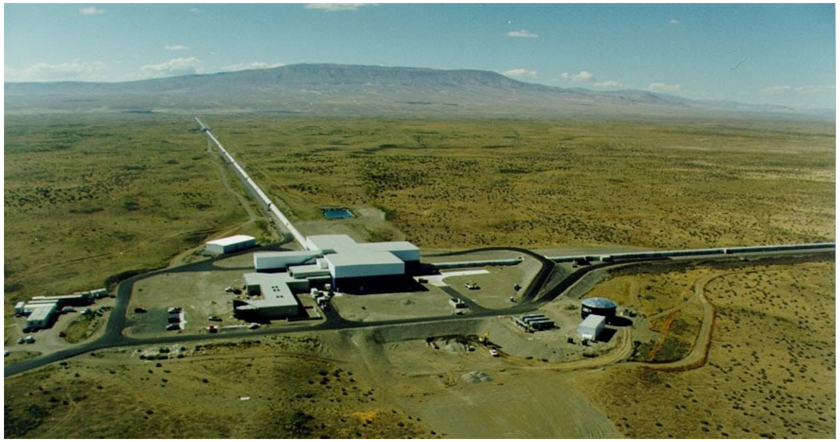
- 14 LRCs with 5 million mappings send Bloom filter updates continuously in Wide Area (unlikely, represents worst case)
- Update times increase when 8 or more clients send updates
- 2 to 3 orders of magnitude better performance than uncompressed (e.g., 5102 seconds with 6 LRCs)

Bloom Filter Compression Supports Higher RLI Query Rates



- Uncompressed updates: about 3000 queries per second
- Higher rates with Bloom filter compression
- Scalability limit: significant overhead to check 100 bit maps
- Practical deployments: <10 LRCs updating an RLI

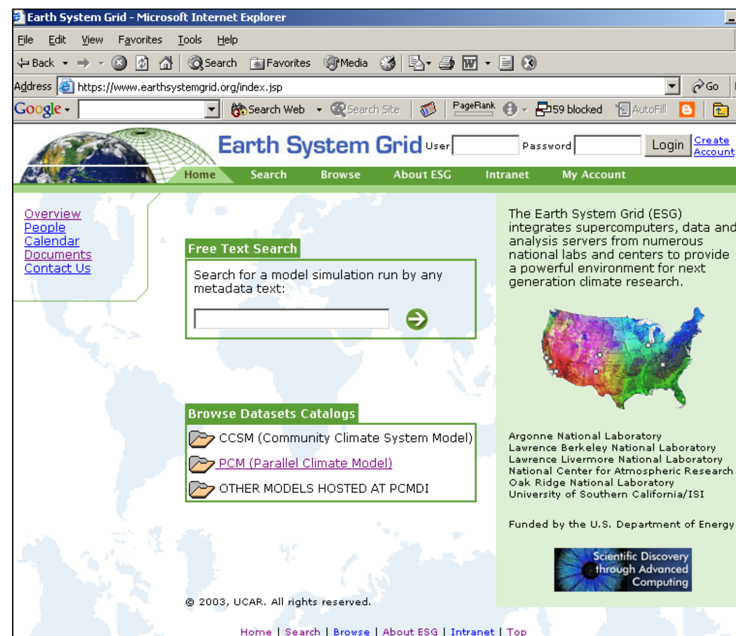
Data Services in Production Use: LIGO



- Laser Interferometer Gravitational Wave Observatory Currently use RLS servers at 10 sites
  - Contain mappings from 6 million logical files to over 40 million physical replicas
- Used in customized data management system: the LIGO Lightweight Data Replicator System (LDR)
  - Includes RLS, GridFTP, custom metadata catalog, tools for storage management and data validation

Data Services in Production Use: ESG

- Earth System Grid: Climate modeling data (CCSM, PCM, IPCC)
- RLS at 4 sites
- Data management coordinated by ESG portal
- Datasets stored at NCAR
  - 64.41 TB in 397253 total files
  - 1230 portal users
- IPCC Data at LLNL
  - 26.50 TB in 59,300 files
  - 400 registered users
  - Data downloaded: 56.80 TB in 263,800 files
  - Avg. 300GB downloaded/day
- (These data are fall 2005)



Data Services in Production Use: Virtual Data System

- Virtual Data System (VDS)
  - Maps from a high-level, abstract definition of a workflow onto a Grid environment
  - Maps to a concrete or executable workflow in the form of a Directed Acyclic Graph (DAG)
  - Passes this concrete workflow to the Condor DAGMan execution system
- VDS uses RLS to
  - Identify physical replicas of logical files specified in the abstract workflow
  - Register new files created during workflow execution

Overview

- Global data services
- Globus building blocks
  - Overview
  - GridFTP
  - Reliable File Transfer Service
  - Replica Location Service
  - **Data Access and Integration Services**
- Building higher-level services
- Application case studies
- Summary

OGSA-DAI: Data Access and Integration for the Grid

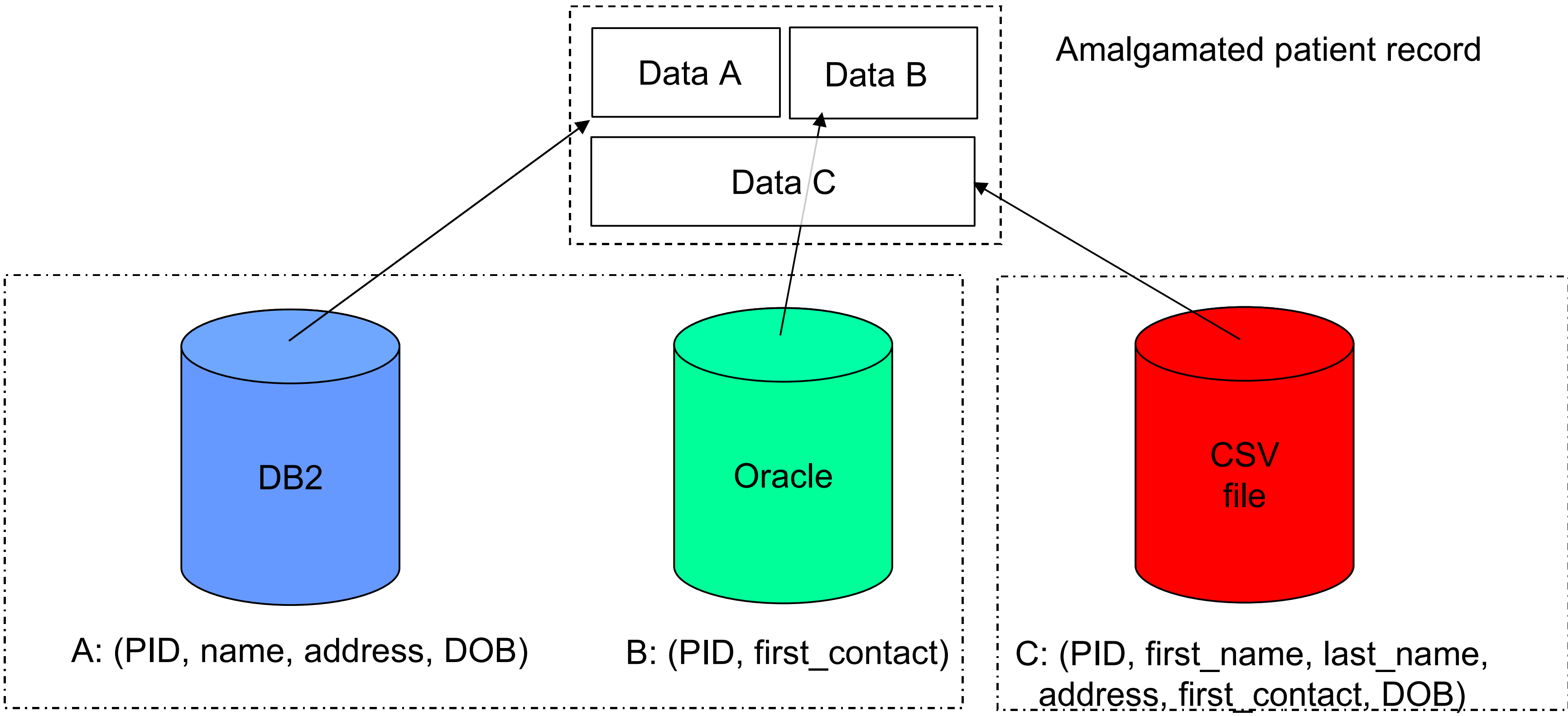
- Focus on structured data (e.g., relational, XML)
- Meet data requirements of Grid applications
  - Functionality, performance and reliability
  - Reduce development cost of data-centric apps
  - Provide consistent interfaces to data resources
- Acceptable and supportable by database providers
  - Trustable, imposed demand is acceptable, etc.
  - Provide a standard framework that satisfies standard requirements

OGSA-DAI Contd.

- A base for developing higher-level services
  - Data federation
  - Distributed query processing
  - Data mining
  - Data visualisation

Integration Scenario

- A patient moves hospital



the globus alliance
www.globus.org

Why OGSA-DAI (and not JDBC)?

- Language independence at the client end
  - Need not use Java
- Platform independence
  - Need not worry about connection technology and drivers
- Can handle XML and file resources
- Can embed additional functionality at the service end
  - Transformations, compression, third party delivery
  - Avoiding unnecessary data movement
- Provision of metadata is powerful
- Usefulness of the registry for service discovery
  - Dynamic service binding process
- The quickest way to make data accessible on the Grid
  - Installation and configuration of OGSA-DAI is fast and straightforward

OGSA-DAI: A Framework for Building Applications

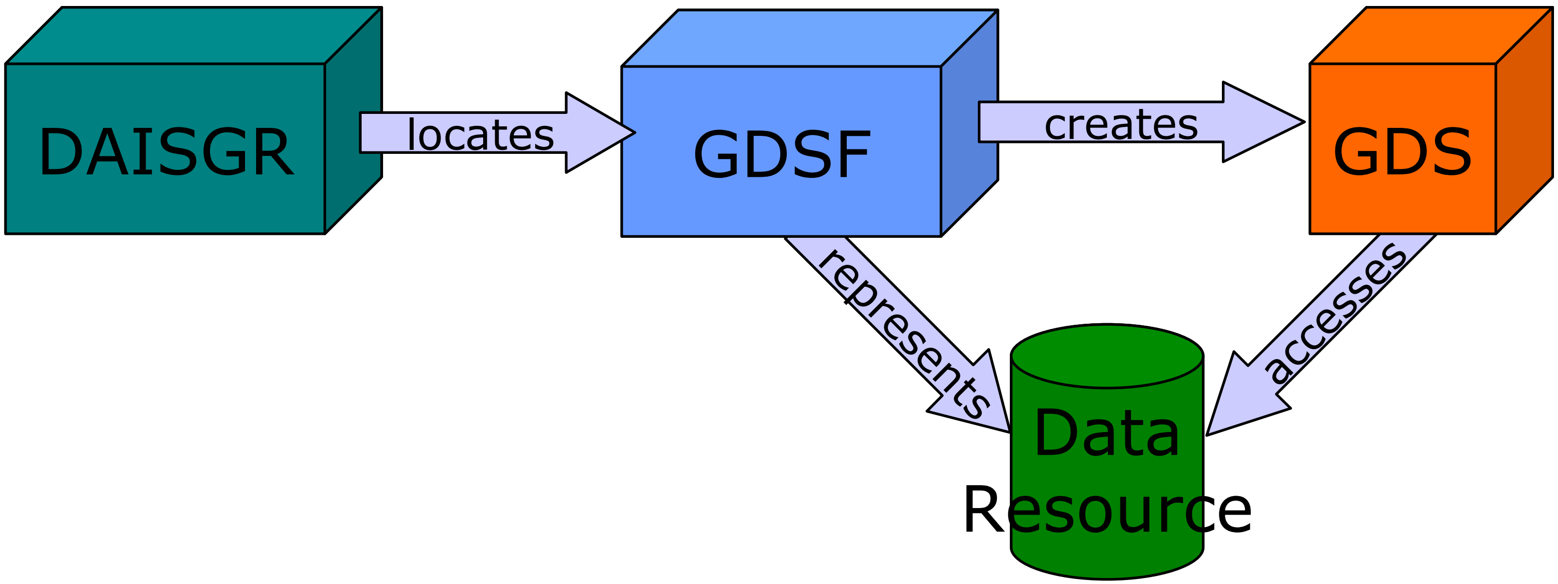
- Supports data access, insert and update
  - MySQL, Oracle, DB2, SQL Server, Postgres
  - XML: Xindice, eXist
  - Files – CSV, BinX, EMBL, OMIM, SWISSPROT,…
- Supports data delivery
  - SOAP over HTTP
  - FTP; GridFTP
  - E-mail
  - Inter-service
- Supports data transformation
  - XSLT
  - ZIP; GZIP

OGSA-DAI: Other Features

- Supports security
  - X.509 certificate-based security
- A framework for building data clients
  - Client toolkit library for app developers
- A framework for developing functionality
  - Extend existing activities, or implement new
  - Mix & match activities to need your needs
- Highly extensible
  - Customise DAIS out-of-the-box product
  - Provide your own services, client-side support, and data-related functionality

OGSA-DAI Services

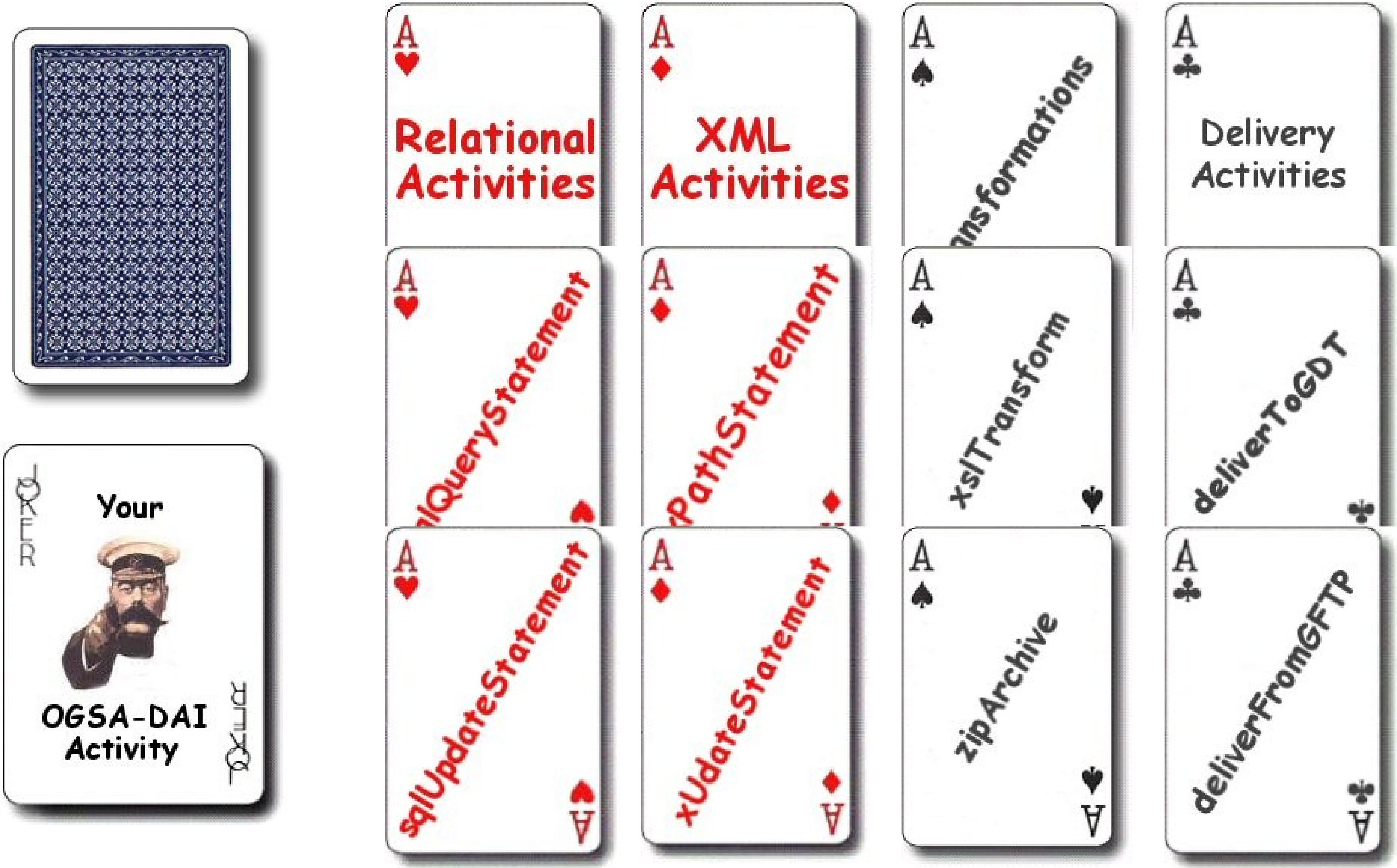
- OGSA-DAI uses three main service types
  - DAISGR (registry) for discovery
  - GDSF (factory) to represent a data resource
  - GDS (data service) to access a data resource



Activities Express Tasks to be Performed by a GDS

- Three broad classes of activities
  - Statement
  - Transformations
  - Delivery
- Extensible
  - Easy to add new functionality
  - No modification to service interface required
  - Extensions operate within OGSA-DAI framework
- Functionality
  - Implemented at the service
  - Work where the data is (need not move data)

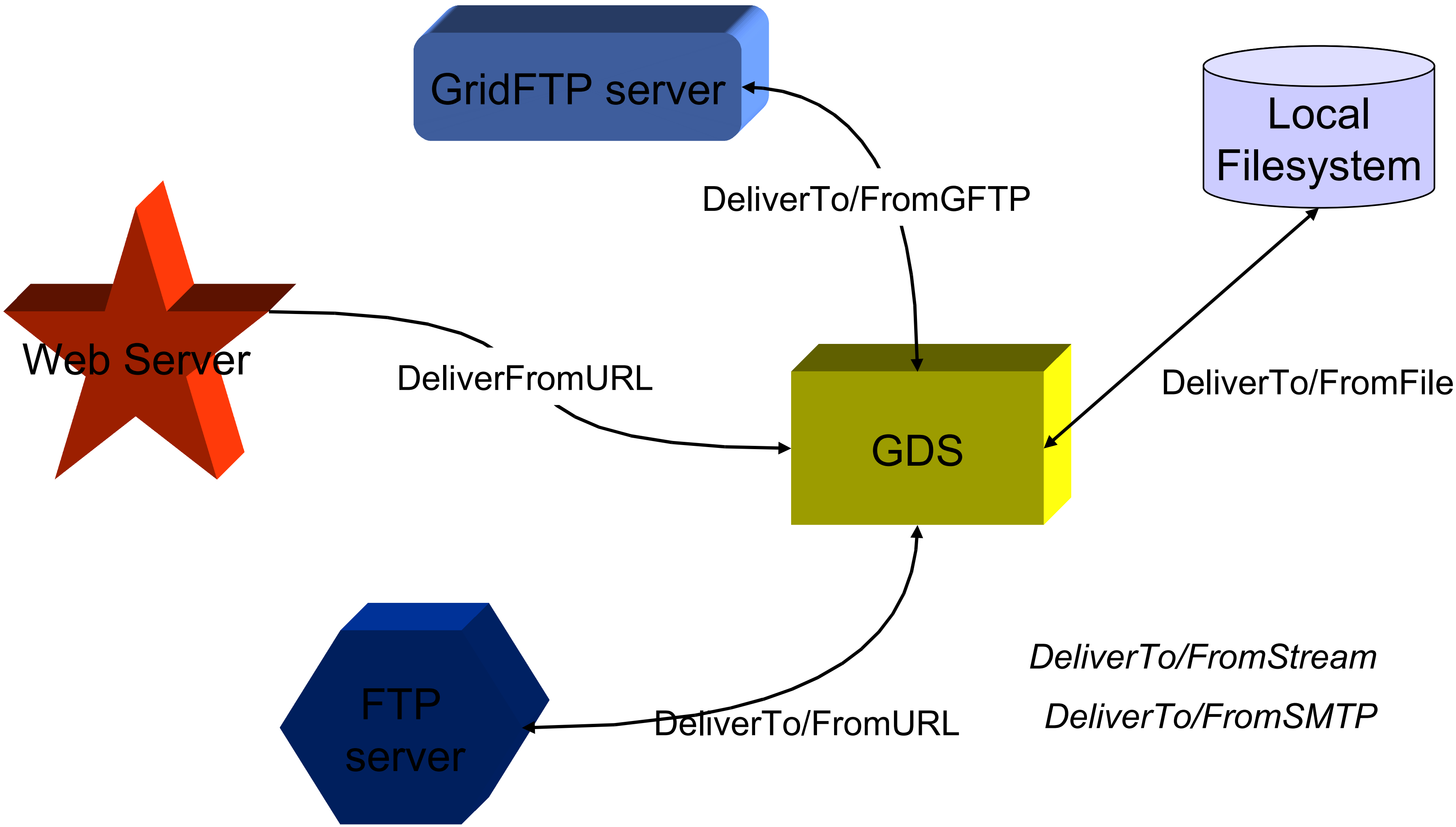
OGSA-DAI Deck



Activities and Requests

- A request contains a set of activities
- An activity dictates an action to be performed
  - Query a data resource
  - Transform data
  - Deliver results
- Data can flow between activities

Delivery Methods



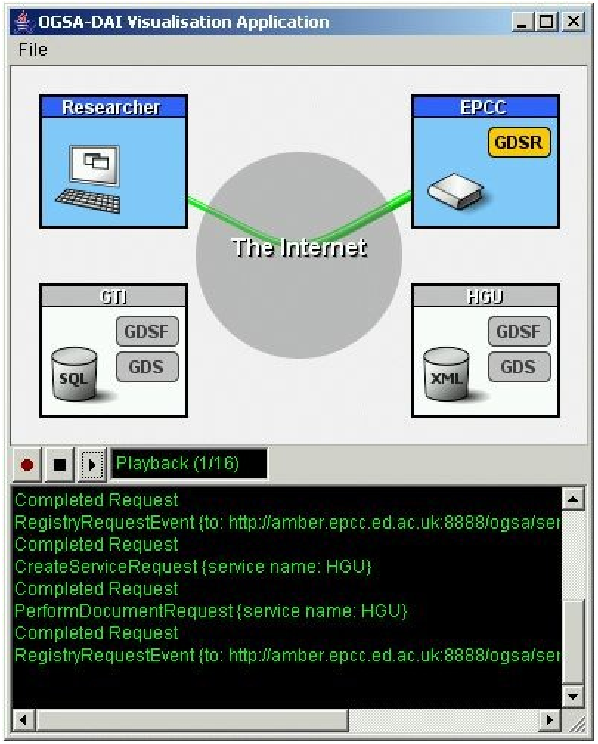
Client Toolkit

- Why? Nobody wants to write XML!
- A programming API which makes writing applications easier
  - Now: Java
  - Next: Perl, C, C#?, ML!?

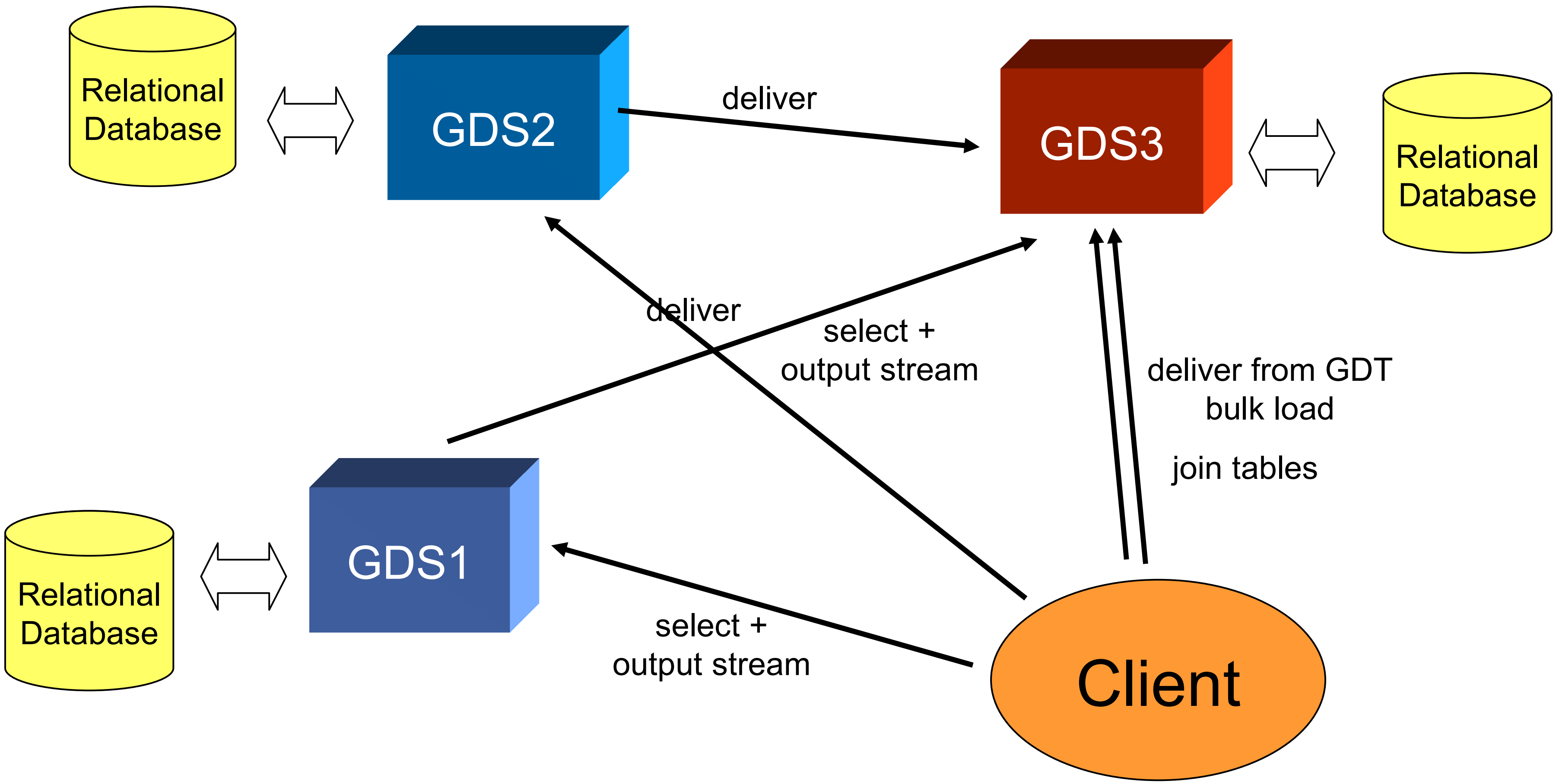
```
// Create a query
SQLQuery query = new SQLQuery(SQLQueryString);
ActivityRequest request = new ActivityRequest();
request.addActivity(query);

// Perform the query
Response response = gds.perform(request);

// Display the result
ResultSet rs = query.getResultSet();
displayResultSet(rs, 1);
```



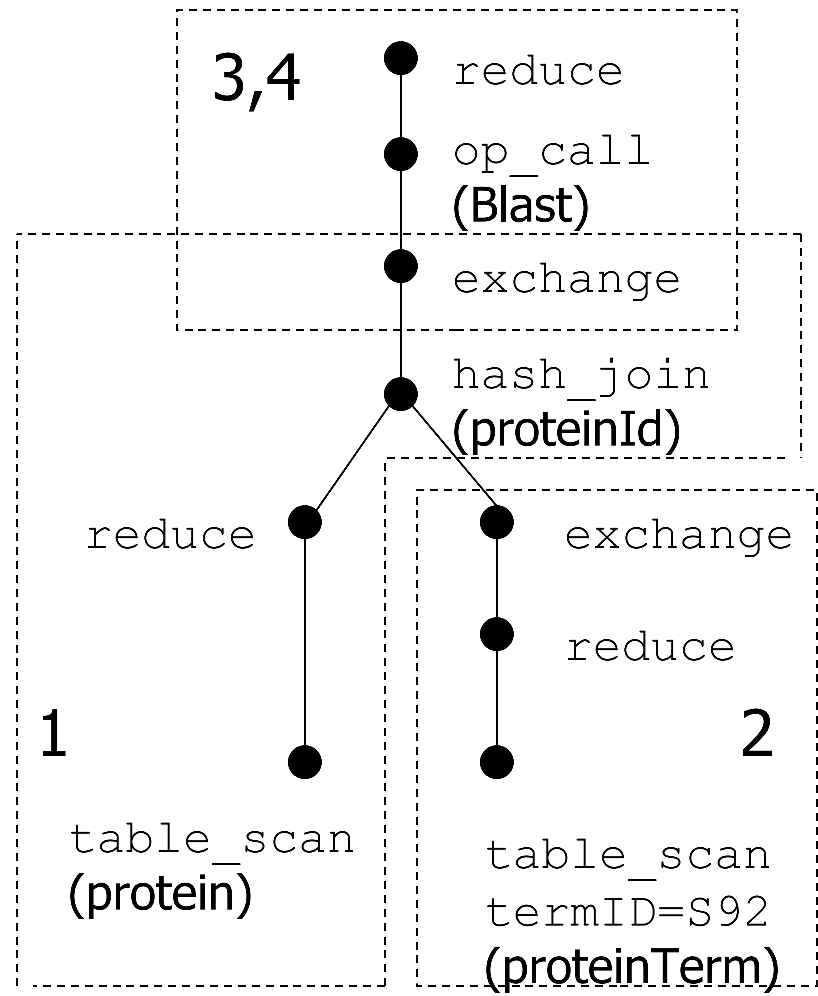
Data Integration Scenario



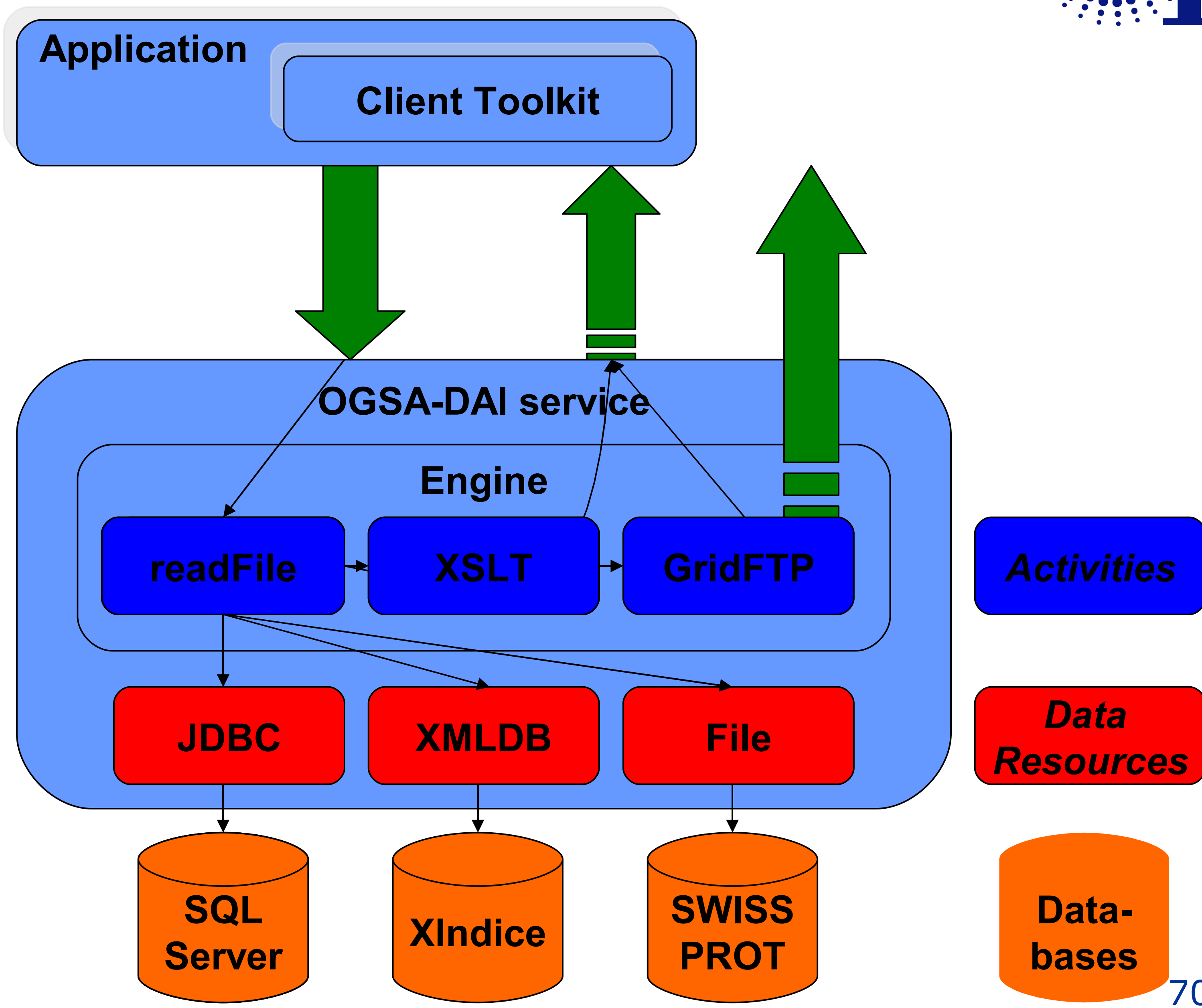
the globus alliance
www.globus.org

Distributed Query Processing

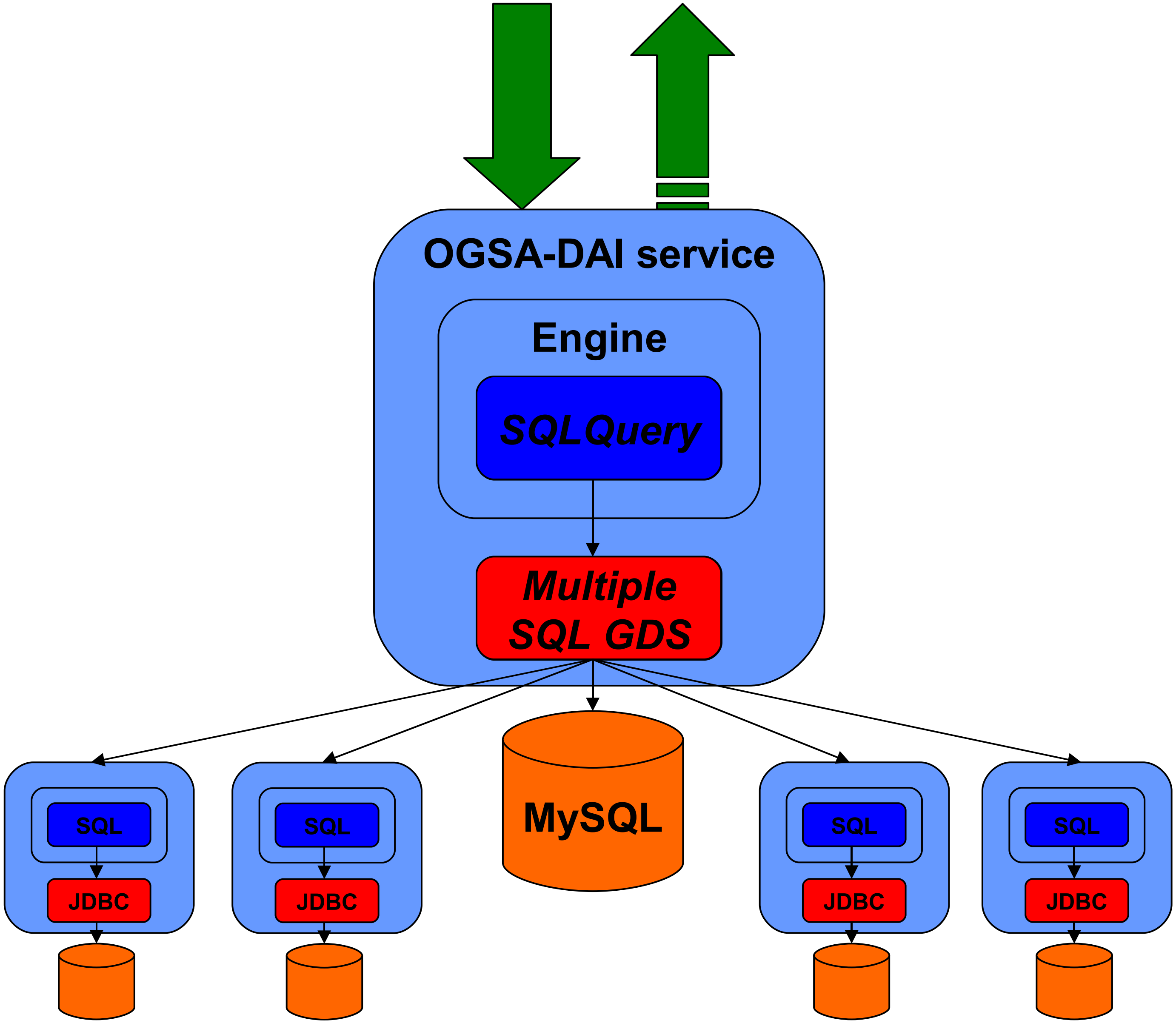
- Higher level services building on OGSA-DAI
- Queries mapped to algebraic expressions for evaluation
- Parallelism represented by partitioning queries
  - Use exchange operators



The OGSA-DAI Framework



Extensibility Example



Overview

- Global data services
- Globus building blocks
- **Building higher-level services**
  - **GRAM execution management service**
  - **Data replication service**
  - **Workflow management**
- Application case studies
- Summary

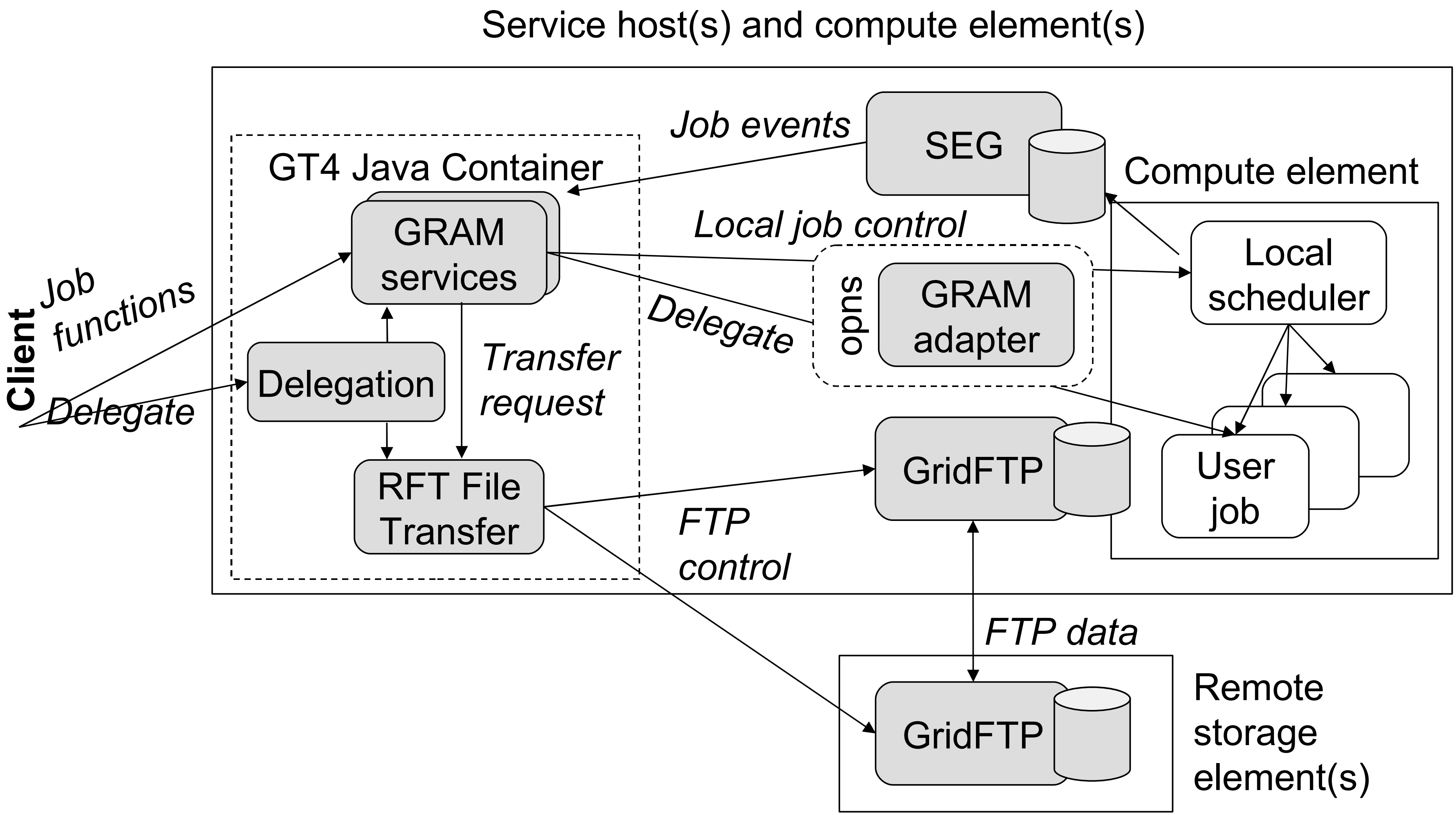
Execution Management (GRAM)

- Common WS interface to schedulers
  - Unix, Condor, LSF, PBS, SGE, …
- More generally: interface for process execution management
  - Lay down execution environment
  - Stage data
  - Monitor & manage lifecycle
  - Kill it, clean up
- A basis for application-driven provisioning

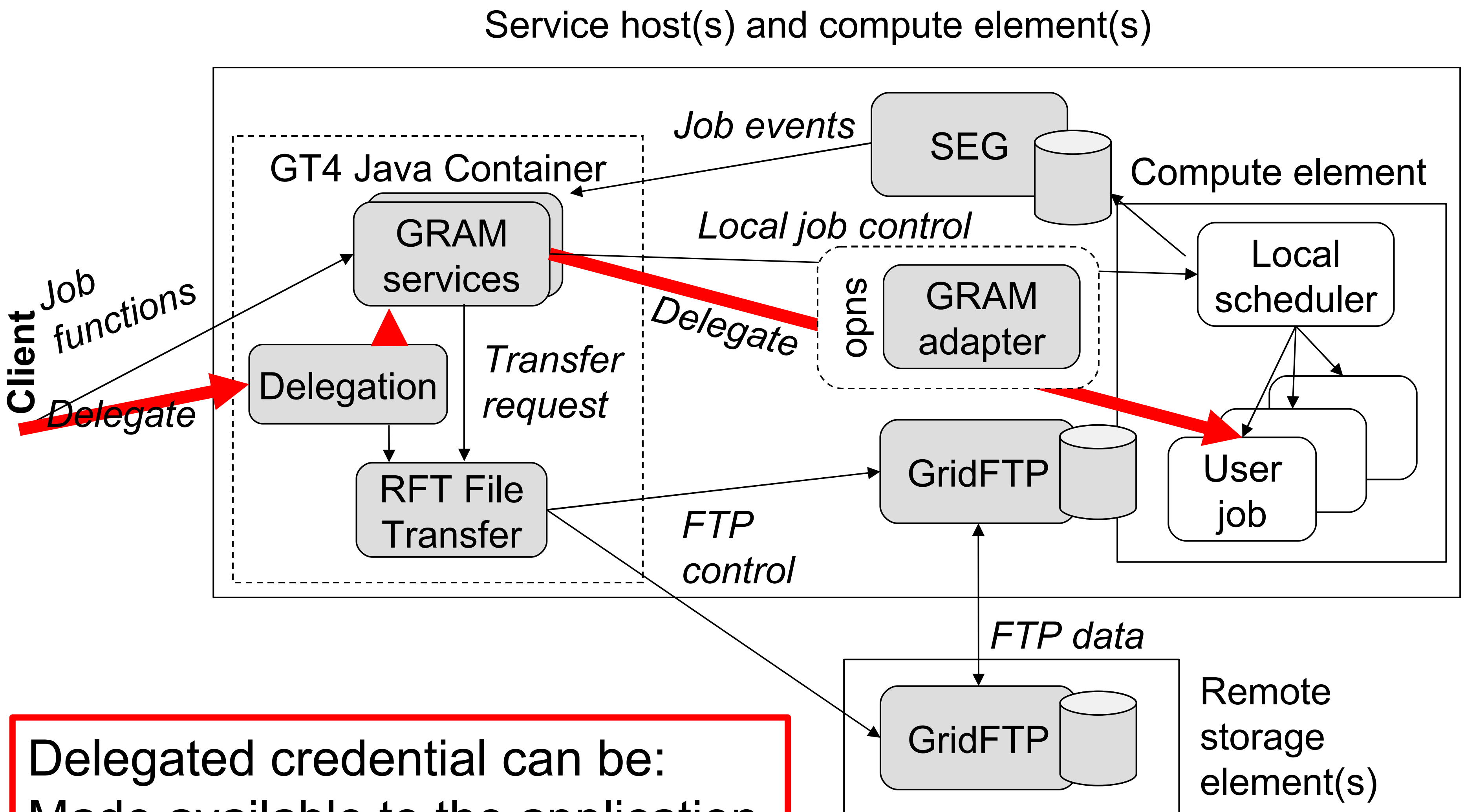
GT4 WS GRAM

- 2nd-generation WS implementation optimized for performance, flexibility, stability, scalability
- Streamlined critical path
  - Use only what you need
- Flexible credential management
  - Credential cache & delegation service
- GridFTP & RFT used for data operations
  - Data staging & streaming output
  - Eliminates redundant GASS code

GT4 WS GRAM Architecture

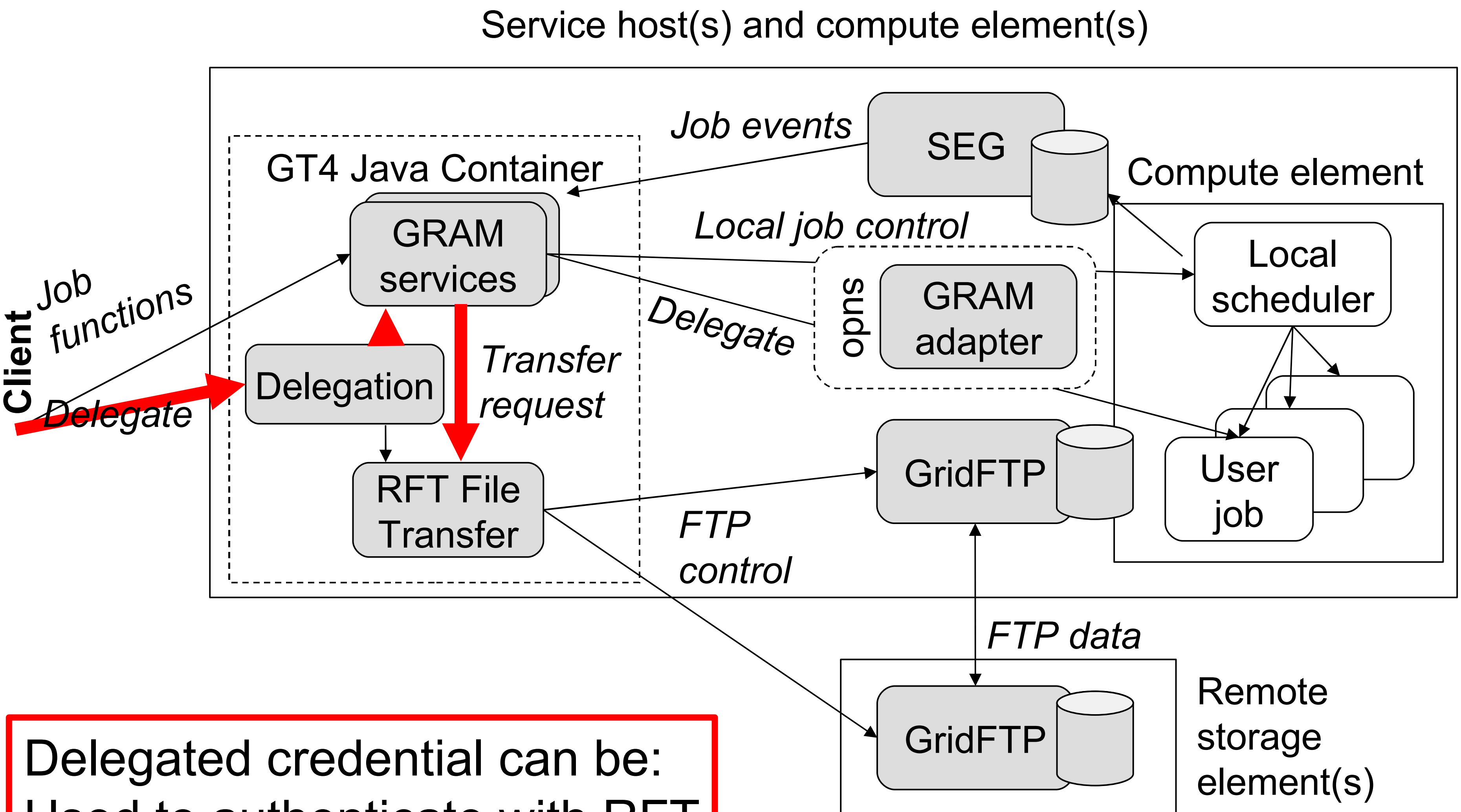


GT4 WS GRAM Architecture



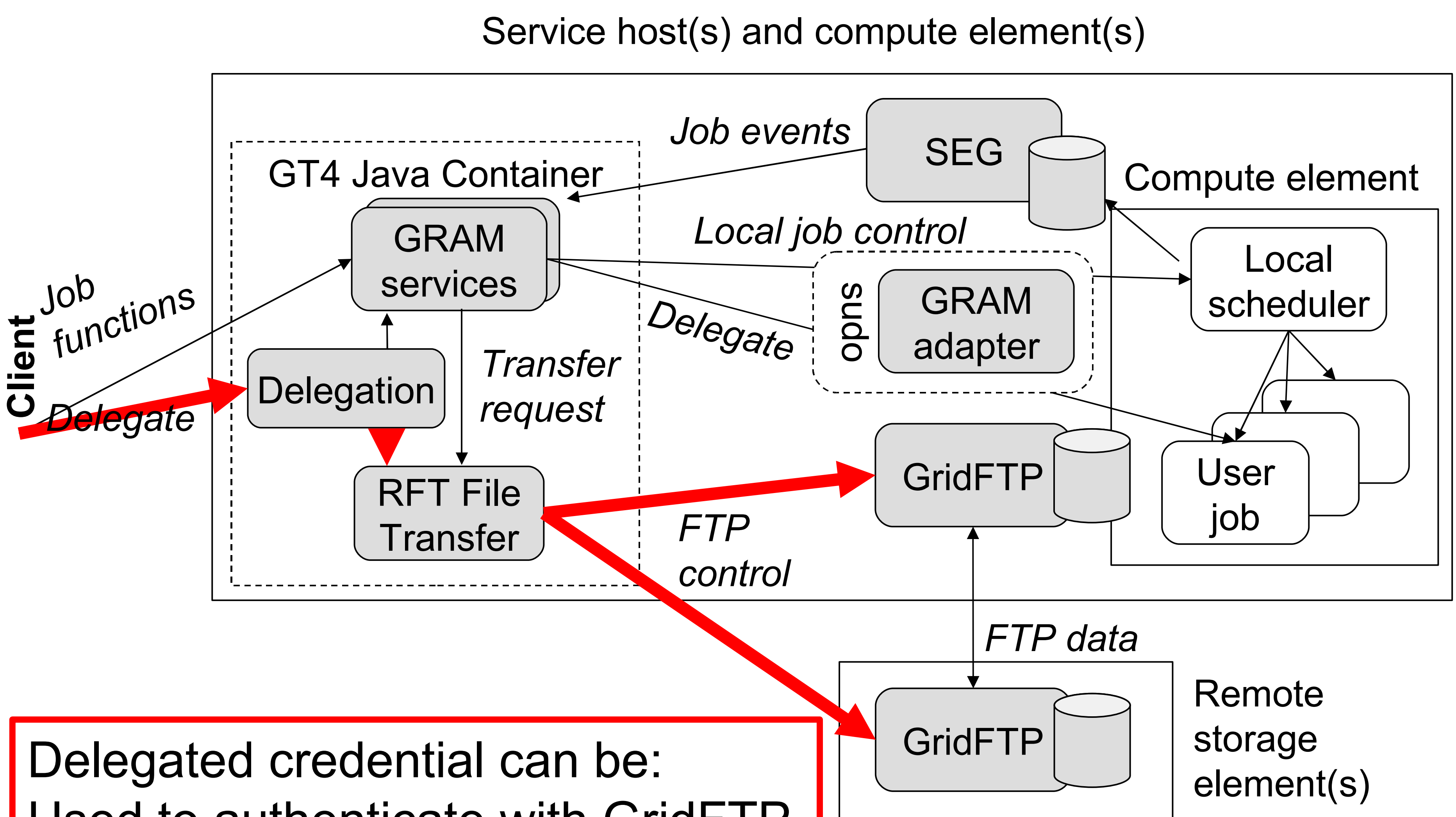
Delegated credential can be:
Made available to the application

GT4 WS GRAM Architecture



Delegated credential can be:
Used to authenticate with RFT

GT4 WS GRAM Architecture



Delegated credential can be:
Used to authenticate with GridFTP

Overview

- Global data services
- Globus building blocks
- Building higher-level services
  - GRAM execution management service
  - **Data replication service**
  - Workflow management
- Application case studies
- Summary

Motivation for Data Replication Services

- Data-intensive applications need higher-level data management services that integrate lower-level Grid functionality
  - Efficient data transfer (GridFTP, RFT)
  - Replica registration and discovery (RLS)
  - Eventually validation of replicas, consistency management, etc.

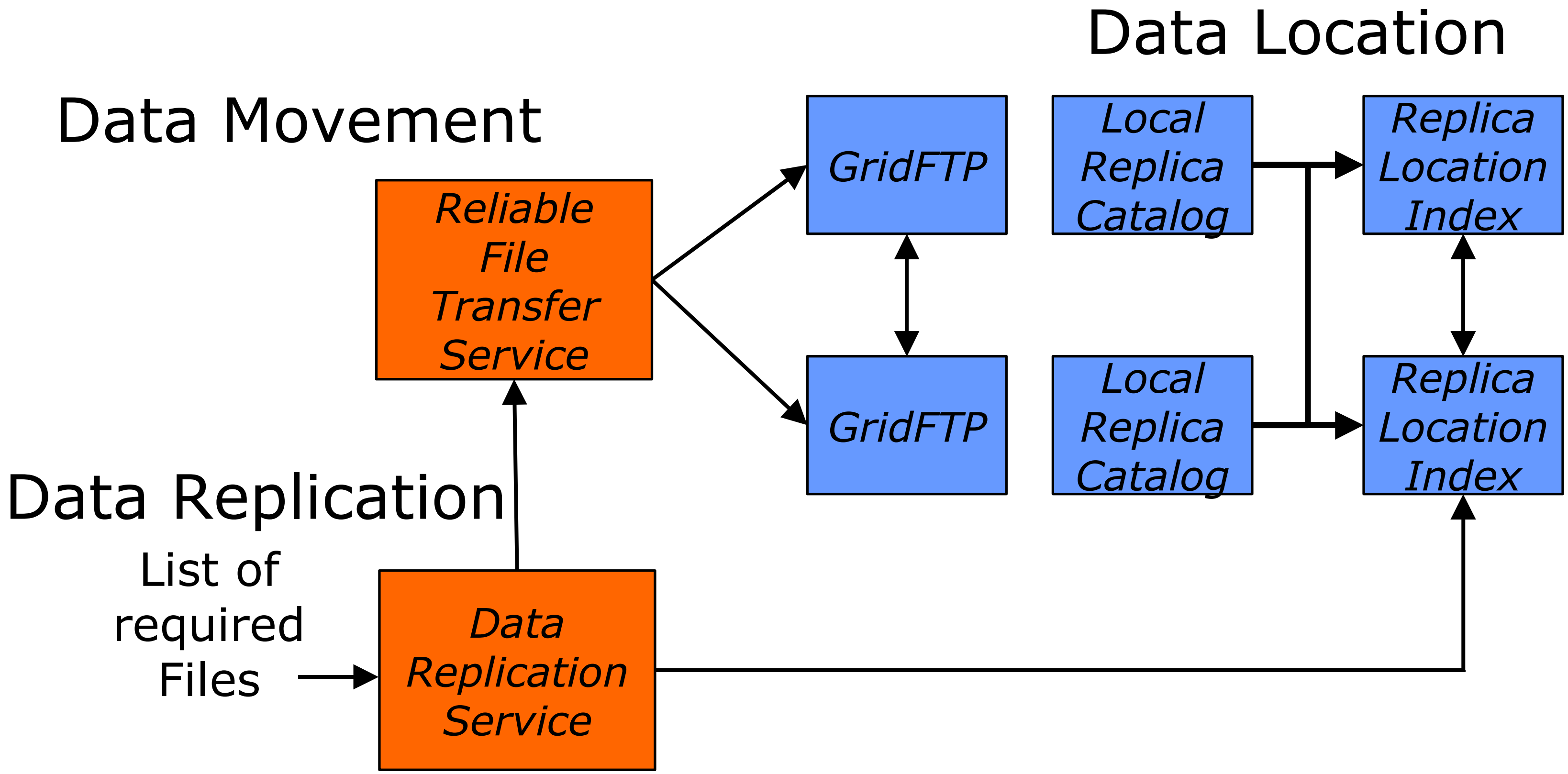
→ Provide a suite of general, configurable, higher-level data management services
  - Data Replication Service is the first of these

Data Replication Service

- Design based on the publication component of the Lightweight Data Replicator system
  - Scott Koranda, U. Wisconsin Milwaukee
- Ensures that specified files exist at a site
  - Compares contents of a local file catalog with a list of desired files
  - Transfers copies of missing files other locations
  - Registers them in the local file catalog
- Uses a pull-based model
  - Localizes decision making; load balancing
  - Minimizes dependency on outside services

Data Replication Service

- Pull "missing" files to a storage system

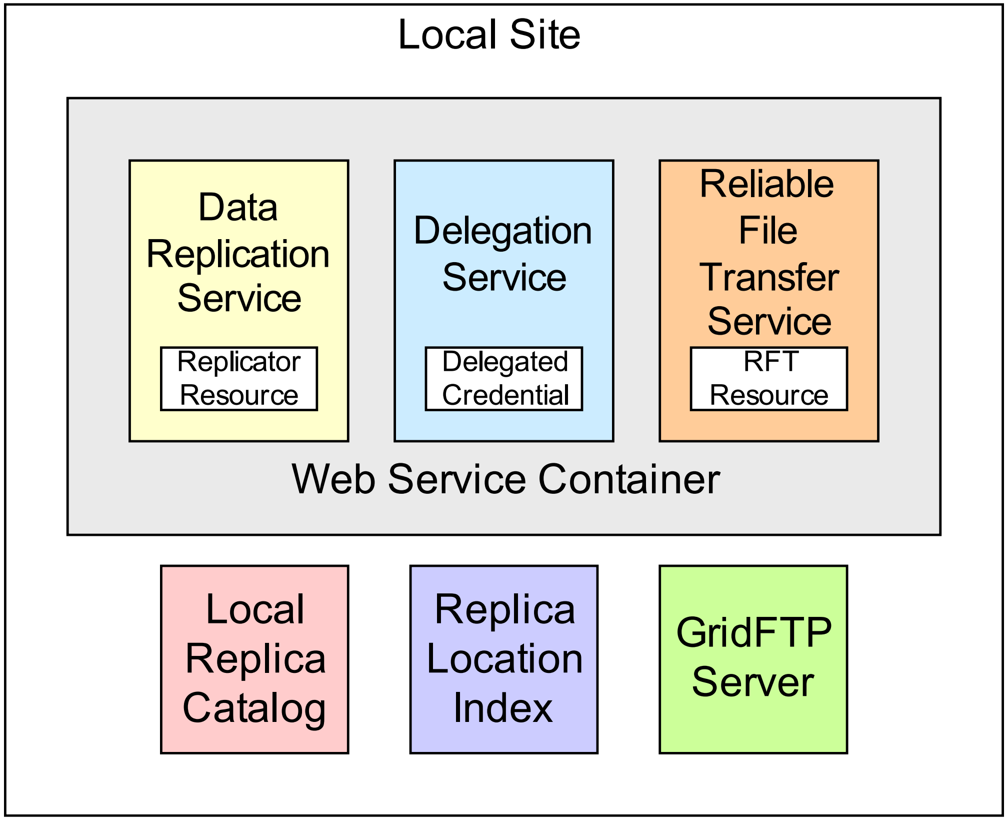


"Design and Implementation of a Data Replication Service Based on the Lightweight Data Replicator System," Chervenak et al., 2005

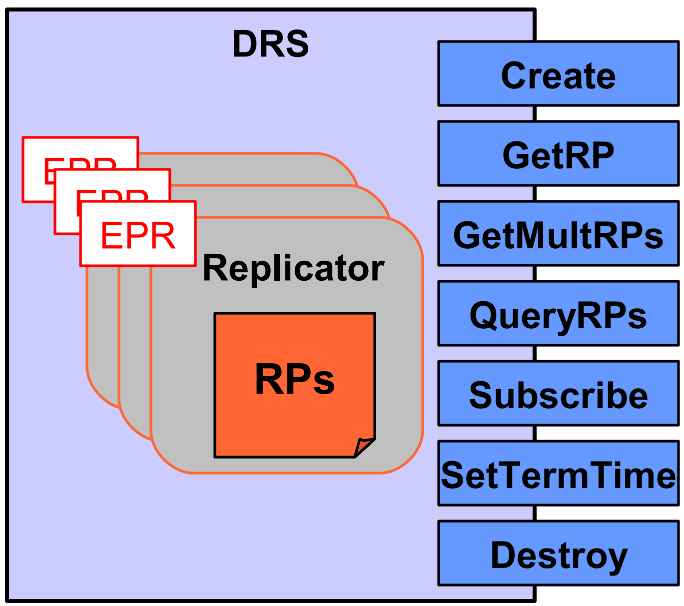
A Typical DRS Deployment

At requesting site, deploy:

- Three Web Services
  - Data Replication Service
  - Delegation Service
  - Reliable File Transfer Service
- Two other services
  - Replica Location Service (Local Replica Catalog, Replica Location Index)
  - GridFTP Server

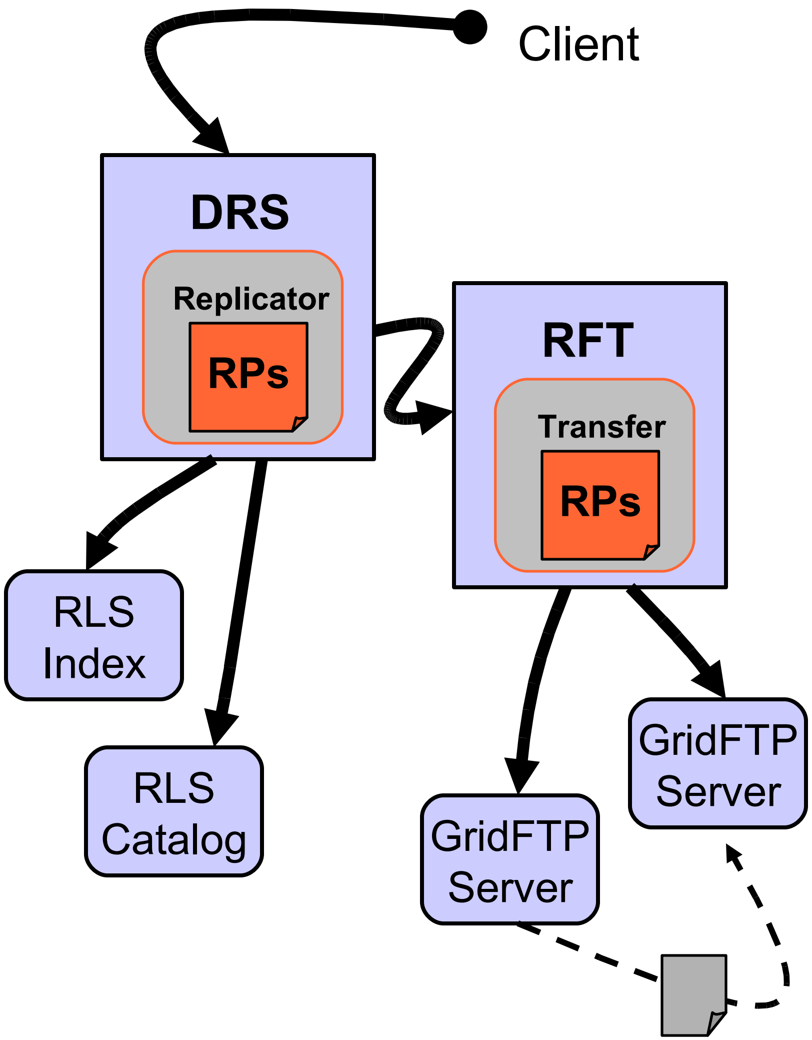


DRS as a WSRF Service



- Web service standards
- CreateReplicator
- State Management
  - Resource
  - Resource Property
- State Identification
  - Endpoint Reference
- Inspection Interfaces
  - GetRP, QueryRPs, GetMultipleRPs
- Notification Interfaces
  - Subscribe
  - Notify
- Lifetime Interfaces
  - SetTerminationTime
  - ImmediateDestruction

DRS Functionality



- **Delegate** credential via Delegation Service
- **Create** a Replicator resource via DRS
- **Discover** replicas of desired files in RLS, **select** among replicas
- **Transfer** data to local site with Reliable File Transfer Service using GridFTP Servers
- **Register** new replicas in RLS catalogs
- **Monitor** Replicator resource and **trigger** events
- **Inspect** state of DRS resource and Resource Properties
- **Destroy** Replicator resource

Performance Measurements: Wide Area Testing

- Destination site for pull-based transfers is Information Sciences Institute (LA)
- Remote site where desired data files are stored is Argonne National Lab (IL)
- DRS operations measured:
  - Create the DRS Replicator resource
  - Discover source files for replication using local RLI and remote LRCs
  - Initiate RFT operation (create RFT resource)
  - Perform RFT data transfer(s)
  - Register the new replicas in the LRC

Experiment 1: Replicate 10 Files of Size 1 Gigabyte

Component of Operation	Time (msec)
Create Replicator Resource	317.0
Discover Files in RLS	449.0
Create RFT Resource	808.6
Transfer Using RFT	1,186,796.0
Register Replicas in RLS	3720.8

- Data transfer time dominates
- Wide area data transfer rate of 67.4 Mbits/sec

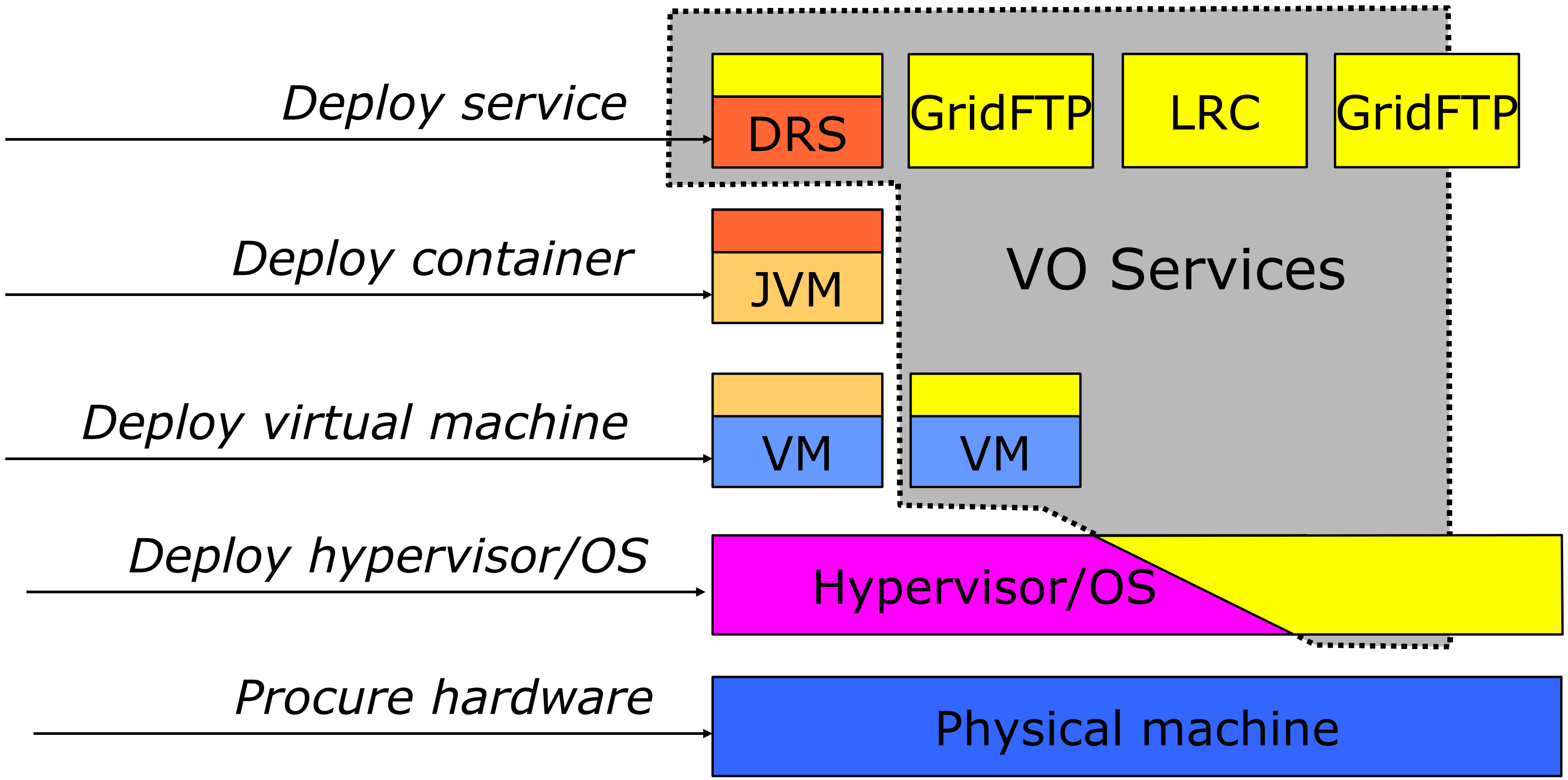
the globus alliance
www.globus.org

Experiment 2: Replicate 1000 Files of Size 10 Megabytes

Component of Operation	Time (msec)
Create Replicator Resource	1561.0
Discover Files in RLS	9.8
Create RFT Resource	1286.6
Transfer Using RFT	963,456.0
Register Replicas in RLS	11,278.2

- Longer to create Replicator and RFT resources
  - Need to store state for 1000 outstanding transfers
- Data transfer time still dominates
- Wide area data transfer rate of 85 Mbits/sec

DRS: Dynamic Deployment



State exposed & access uniformly at all levels
Provisioning, management, and monitoring at all levels

Overview

- Global data services
- Globus building blocks
- Building higher-level services
  - GRAM execution management service
  - Data replication service
  - **Workflow management**
- Application case studies
- Summary

Data-Intensive Workflow (www.griphyn.org)

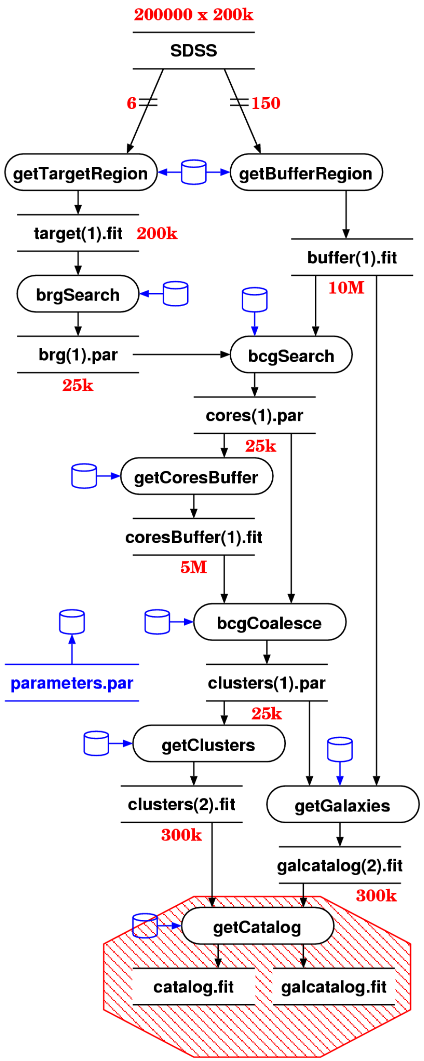
Enhance scientific productivity through…

- Discovery, application and management of data and processes at petabyte scale
- Using a worldwide data grid as a scientific workstation

The key to this approach is Virtual Data – creating and managing datasets through workflow "recipes" and provenance recording.

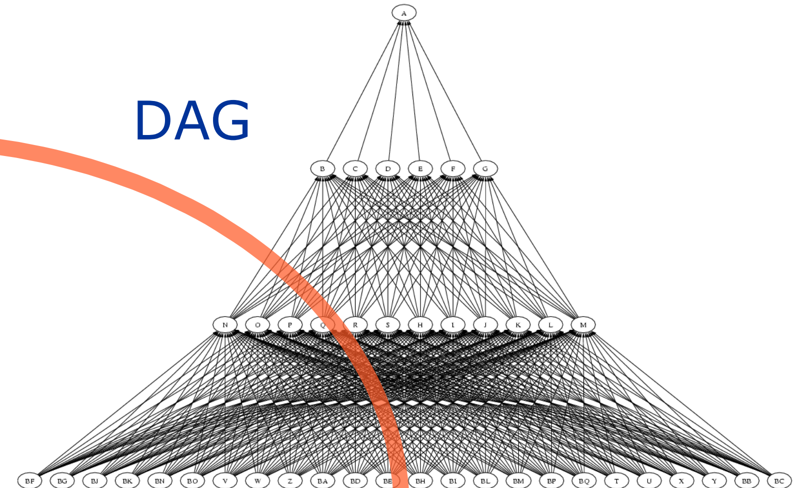
the globus alliance
www.globus.org

Virtual Data Example: *Galaxy Cluster Search*

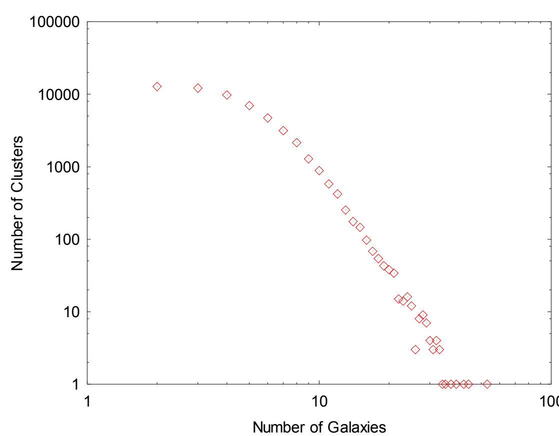


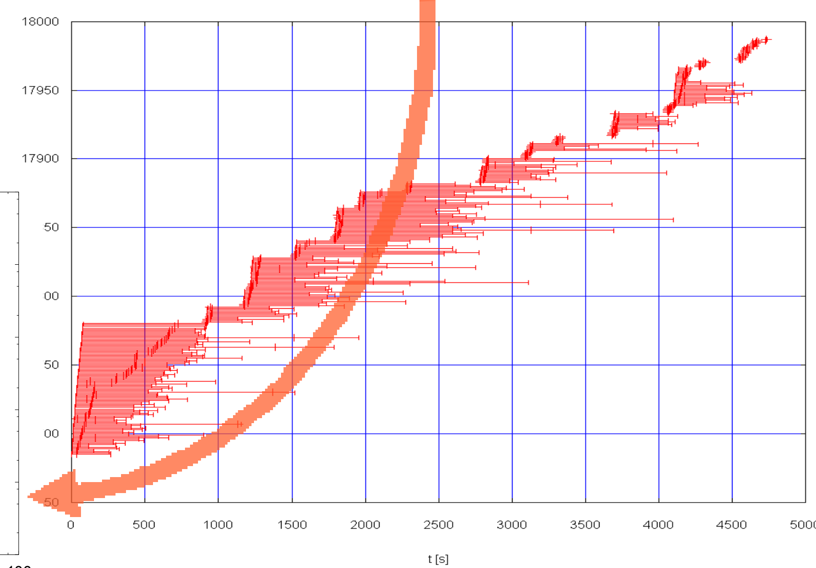
Sloan Data

DAG



Galaxy cluster size distribution



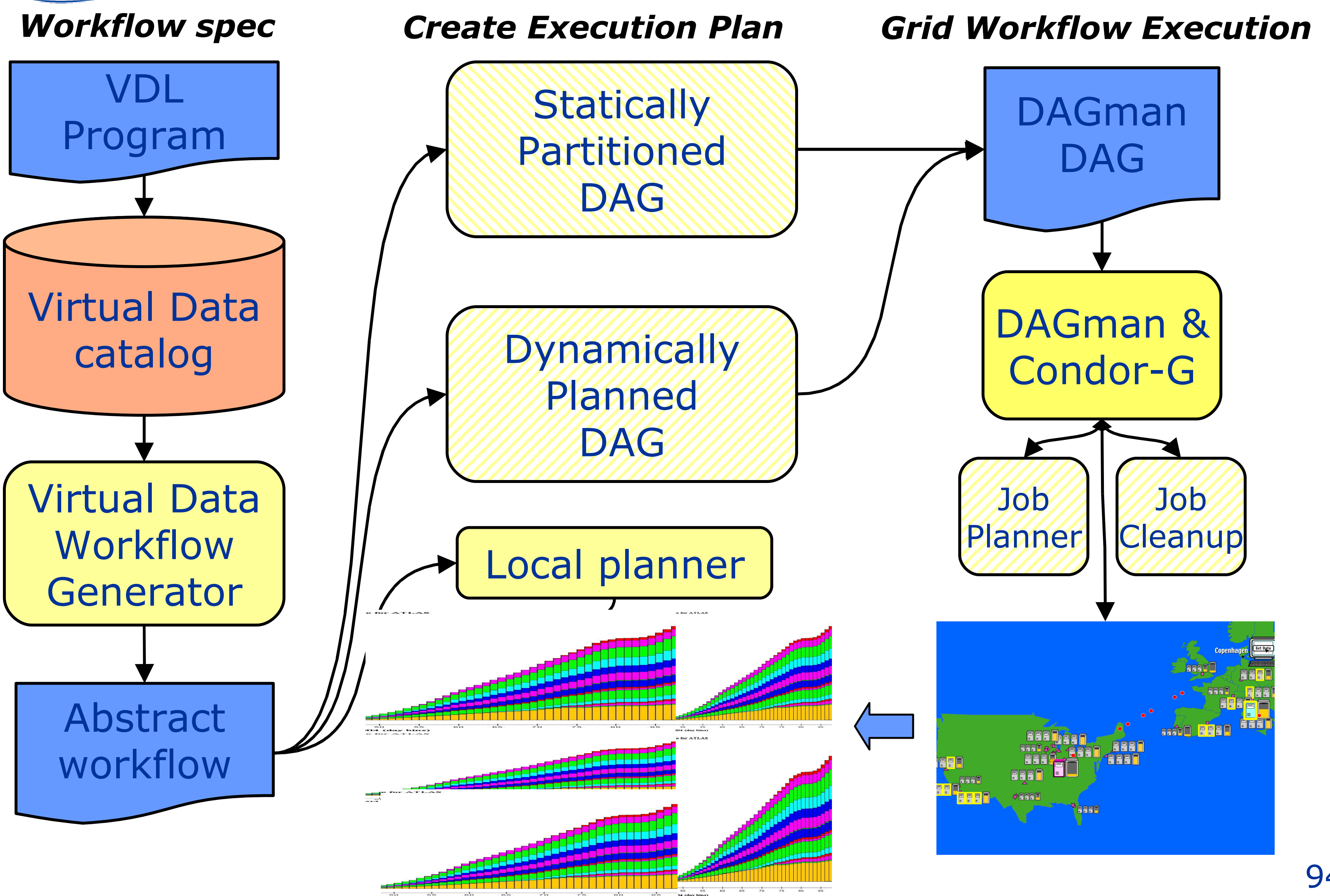


Jim Annis, Steve Kent, Vijay Sehkri, *Fermilab*, Michael Milligan, Yong Zhao, *University of Chicago*

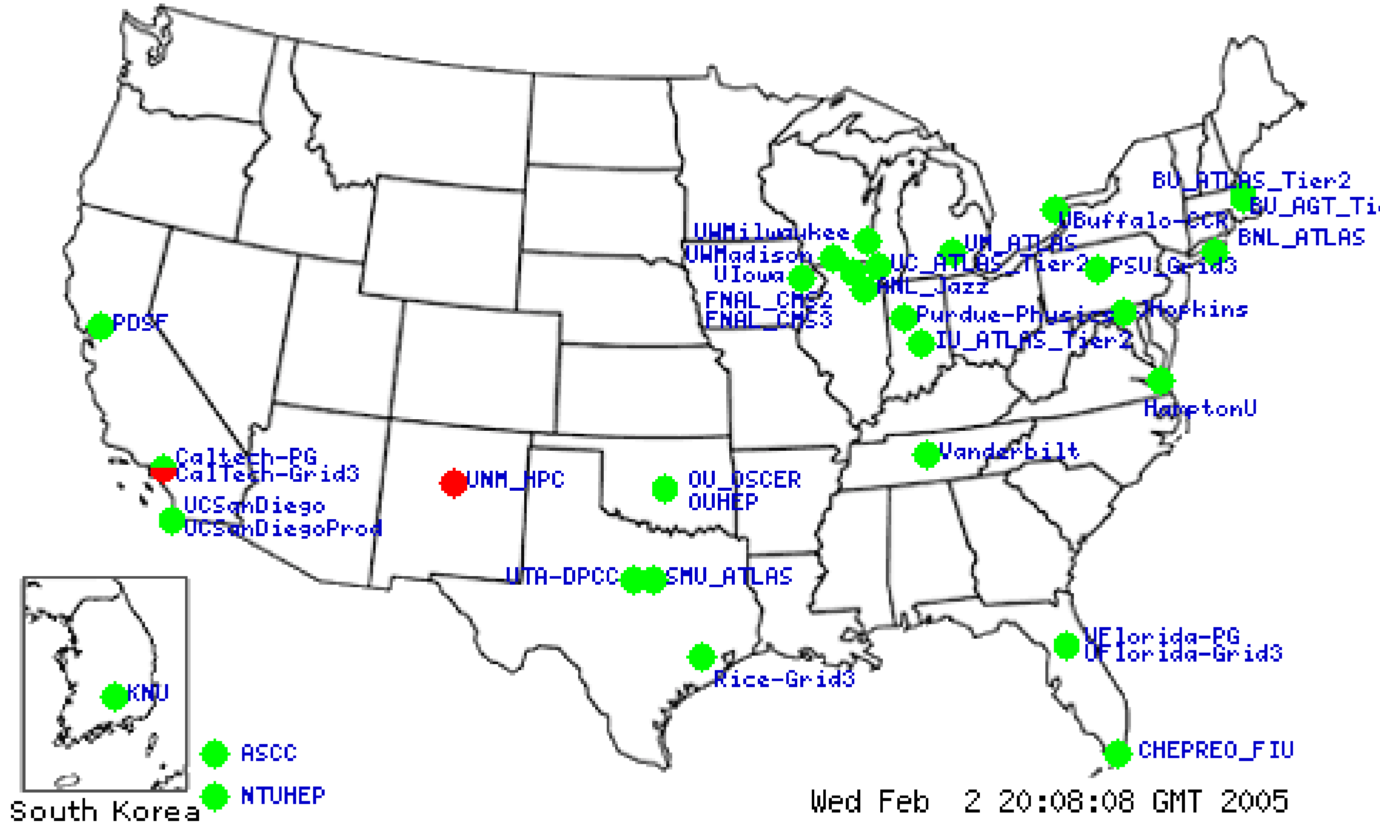
What Must we "Virtualize" to Compute on the Grid?

- Location-independent computing: represent all workflow in abstract terms
- Declarations not tied to specific entities:
  - Sites
  - File systems
  - Schedulers
- Failures – automated retry for data server and execution site un-availability

Executing VDS Workflows



OSG:The “target chip” for VDS Workflows



Supported by the National Science Foundation and the Department of Energy.

the globus alliance
www.globus.org

VDS Applications

Application	Jobs / workflow	Levels	Status
ATLAS HEP Event Simulation	500K	1	In Use
LIGO Inspiral/Pulsar	~700	2-5	Inspiral In Use
NVO/NASA Montage/Morphology	1000s	7	Both In Use
GADU Genomics: BLAST,…	40K	1	In Use
fMRI DBIC AIRSN Image Proc	100s	12	In Devel
QuarkNet CosmicRay science	<10	3-6	In Use
SDSS Coadd; Cluster Search	40K 500K	2 8	In Devel / CS Research
FOAM Ocean/Atmos Model	2000 (core app runs 250 8-CPU jobs)	3	In use
GTOMO Image proc	1000s	1	In Devel
SCEC Earthquake sim	1000s		In use

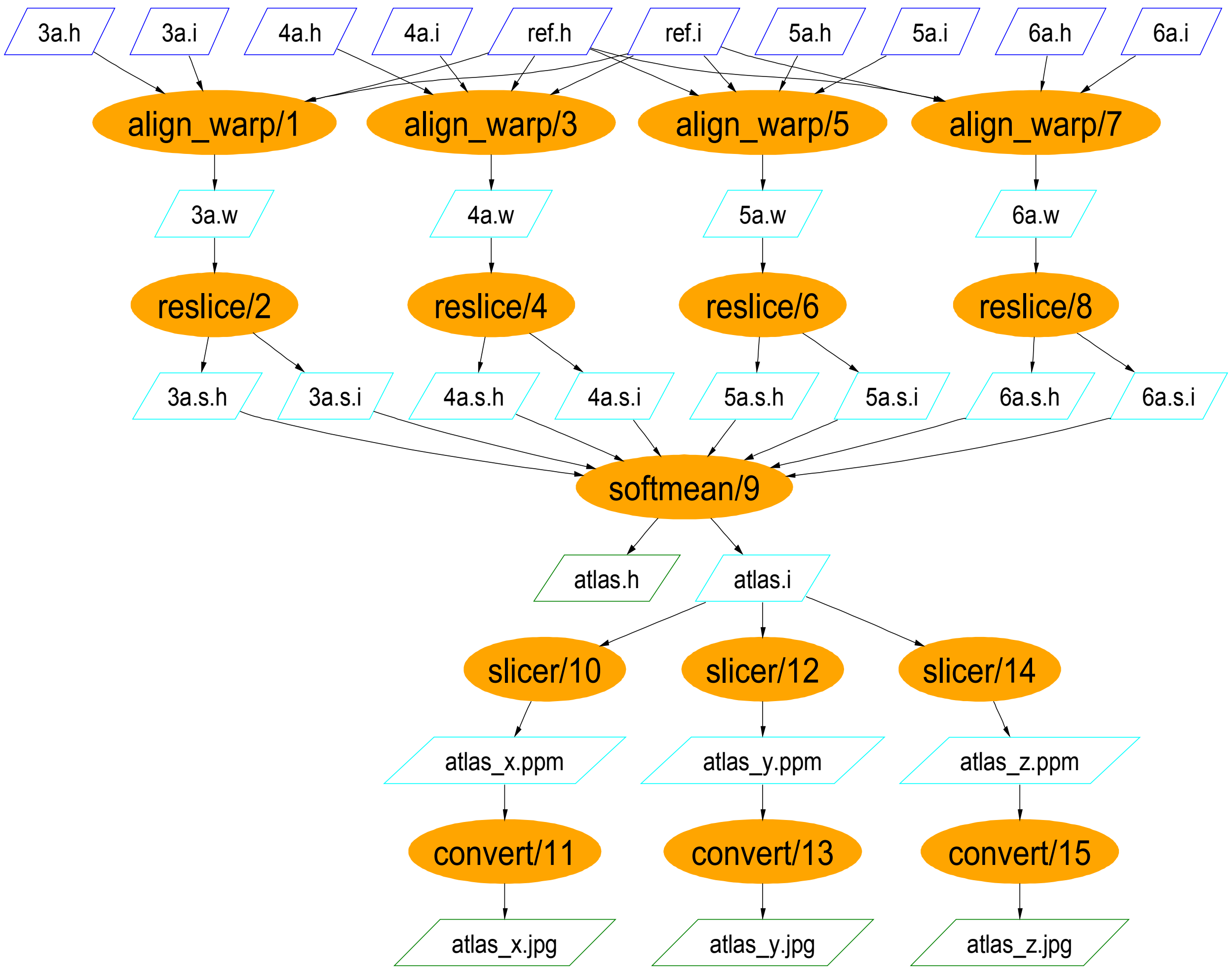
A Case Study – Functional MRI

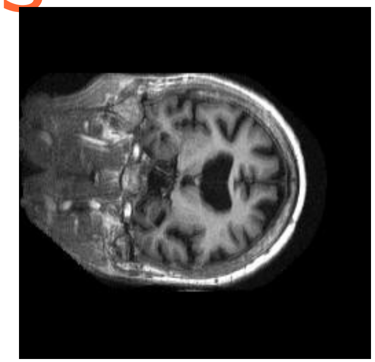
- Problem: “spatial normalization” of a images to prepare data from fMRI studies for analysis
- Target community is approximately 60 users at Dartmouth Brain Imaging Center
- Wish to share data and methods across country with researchers at Berkeley
- Process data from arbitrary user and archival directories in the center’s AFS space; bring data back to same directories
- Grid needs to be transparent to the users: Literally, “Grid as a Workstation”

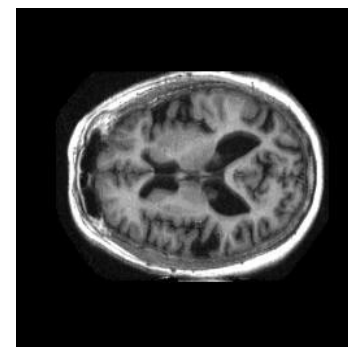
A Case Study – Functional MRI (2)

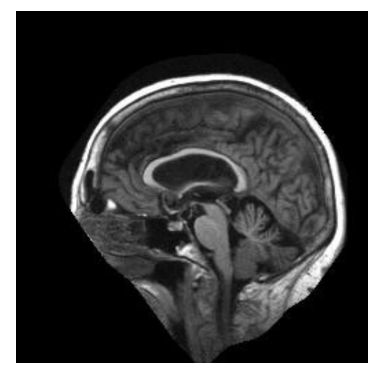
- Based workflow on shell script that performs 12-stage process on a local workstation
- Adopted replica naming convention for moving user's data to Grid sites
- Creates VDL pre-processor to iterate transformations over datasets
- Using resources across two distinct grids – OSG and Dartmouth Green Grid

Functional MRI Analysis



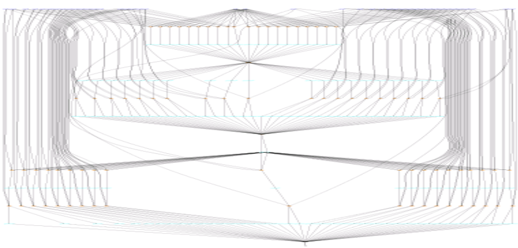


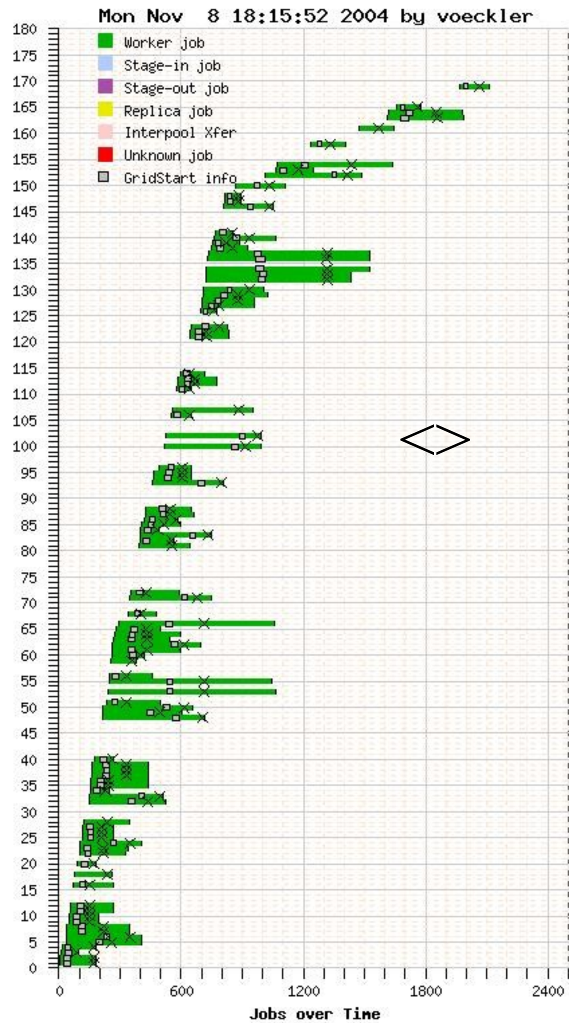


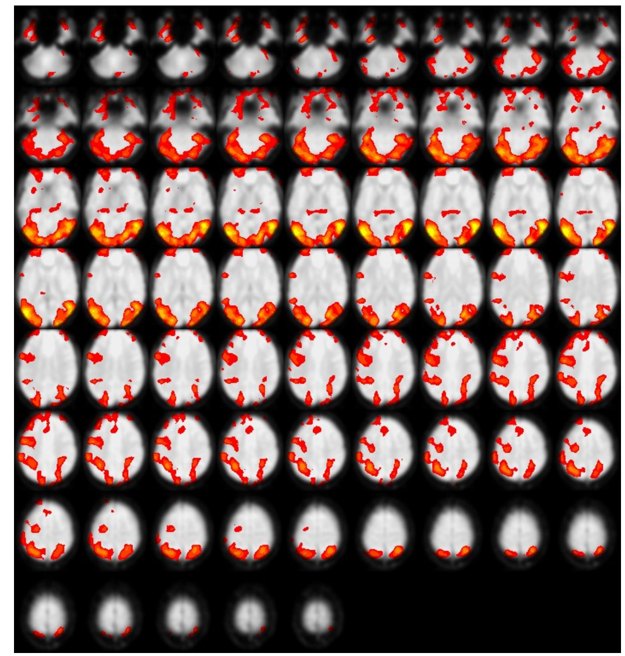


Workflow courtesy James Dobson, Dartmouth Brain Imaging Center

Functional MRI – Mapping Brain Function using Grid Workflows







fMRI Virtual Data Queries

Which transformations can process a "subject image"?

- Q: xsearchvdc -q tr_meta dataType subject_image input
- A: fMRIDC.AIR::align_warp

List anonymized subject-images for young subjects:

- Q: xsearchvdc -q lfn_meta dataType subject_image privacy anonymized subjectType young
- A: 3472-4_anonymized.img

Show files that were derived from patient image 3472-3:

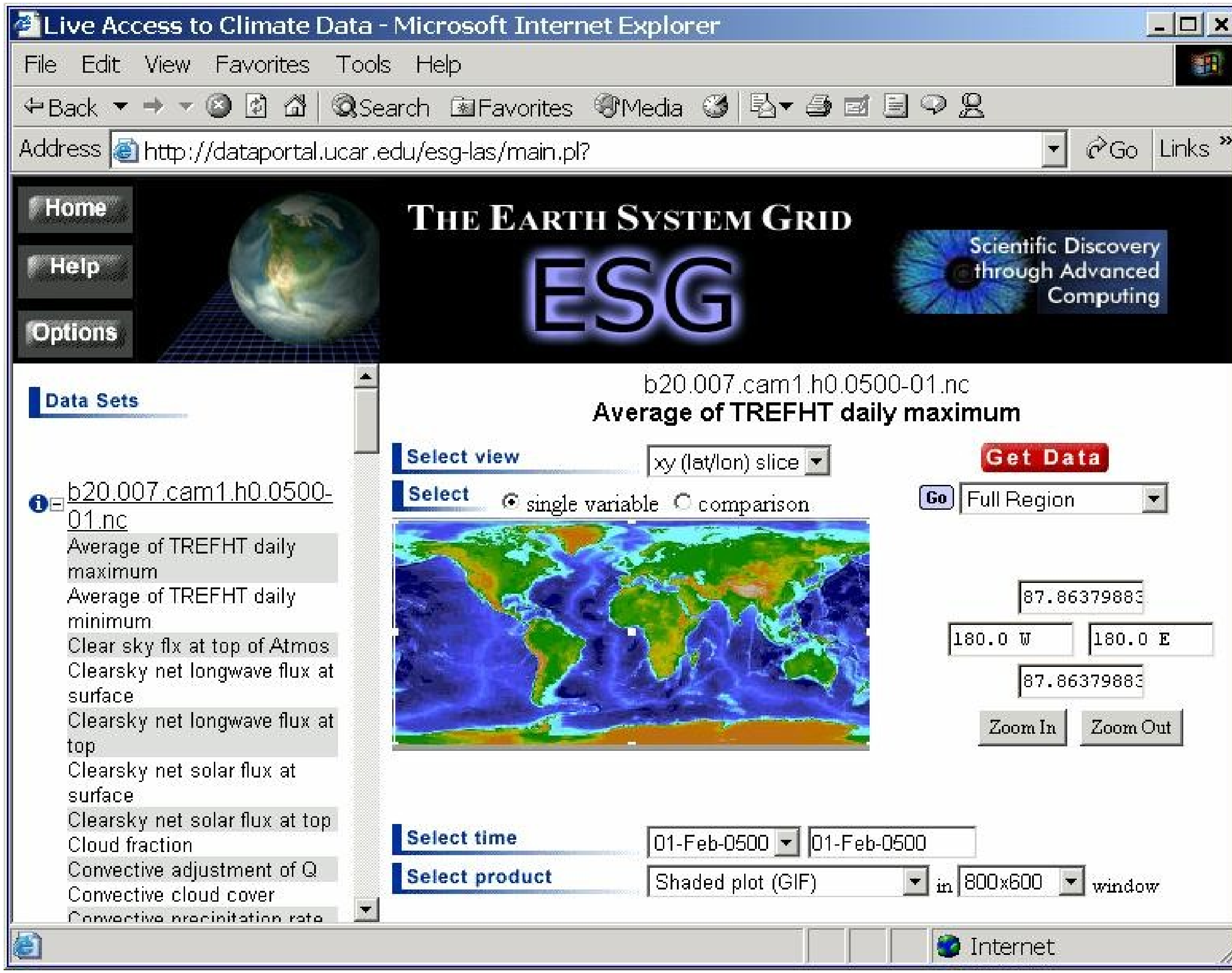
- Q: xsearchvdc -q lfn_tree 3472-3_anonymized.img
- A: 3472-3_anonymized.img
  3472-3_anonymized.sliced.hdr
  atlas.hdr
  atlas.img
  …
  atlas_z.jpg
  3472-3_anonymized.sliced.img

Overview

- Global data services
- Building blocks
- **Case studies**
  - **Earth System Grid**
  - **Southern California Earthquake Center**
  - **Cancer Bioinformatics Grid**
  - **AstroPortal stacking service**
  - **GADU bioinformatics service**
- Summary

Earth System Grid

Goal: address technical obstacles to the sharing & analysis of high-volume data from advanced earth system models



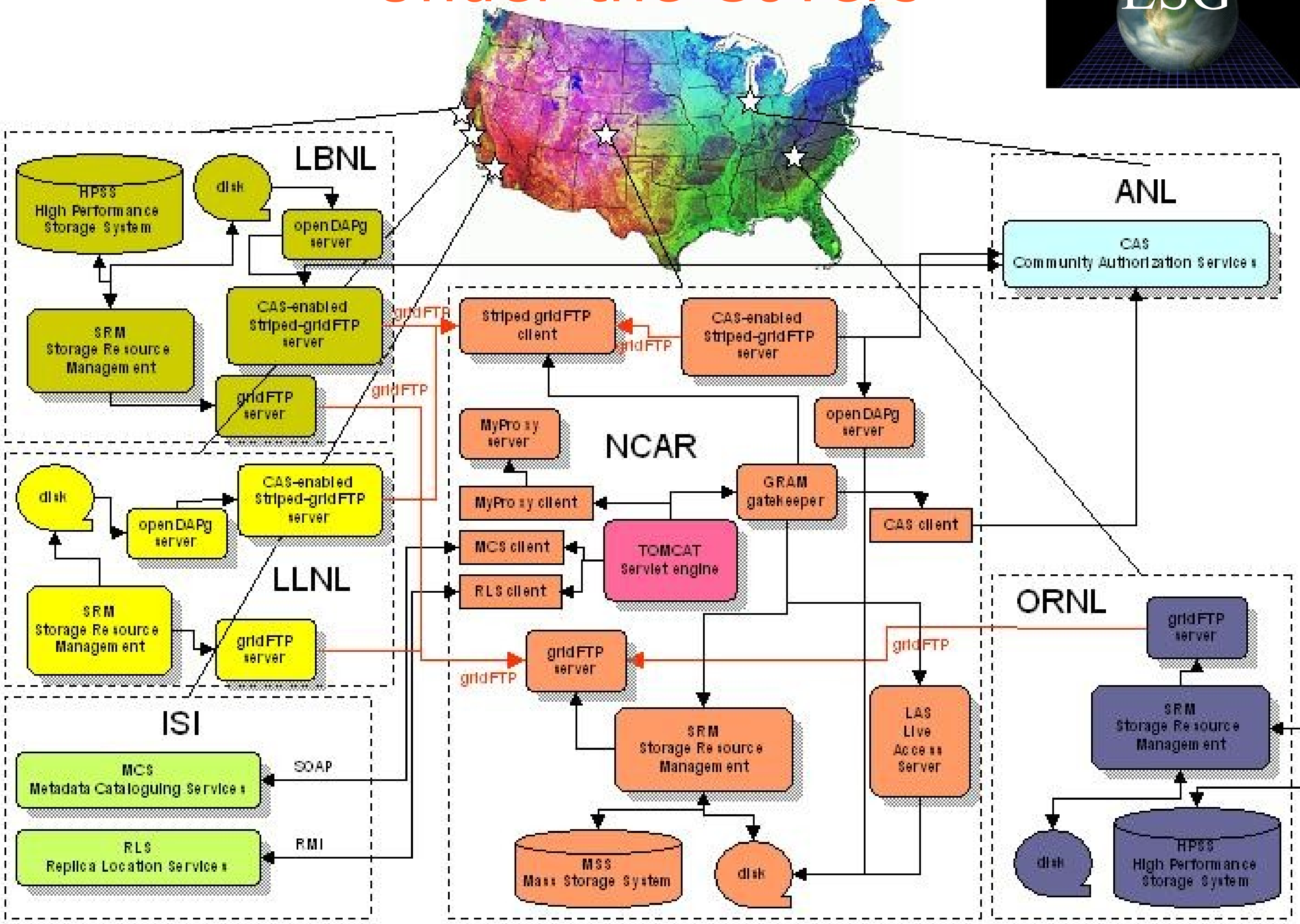
ESG Requirements

- Move data a minimal amount, keep it close to computational point of origin when possible
- When we must move data, do it fast and with minimum human intervention
- Keep track of what we have, particularly what's on deep storage
- Make use of the facilities available at multiple sites (centralization not an option)
- Data must be easy to find and access using standard Web browsers

Major ESG Components

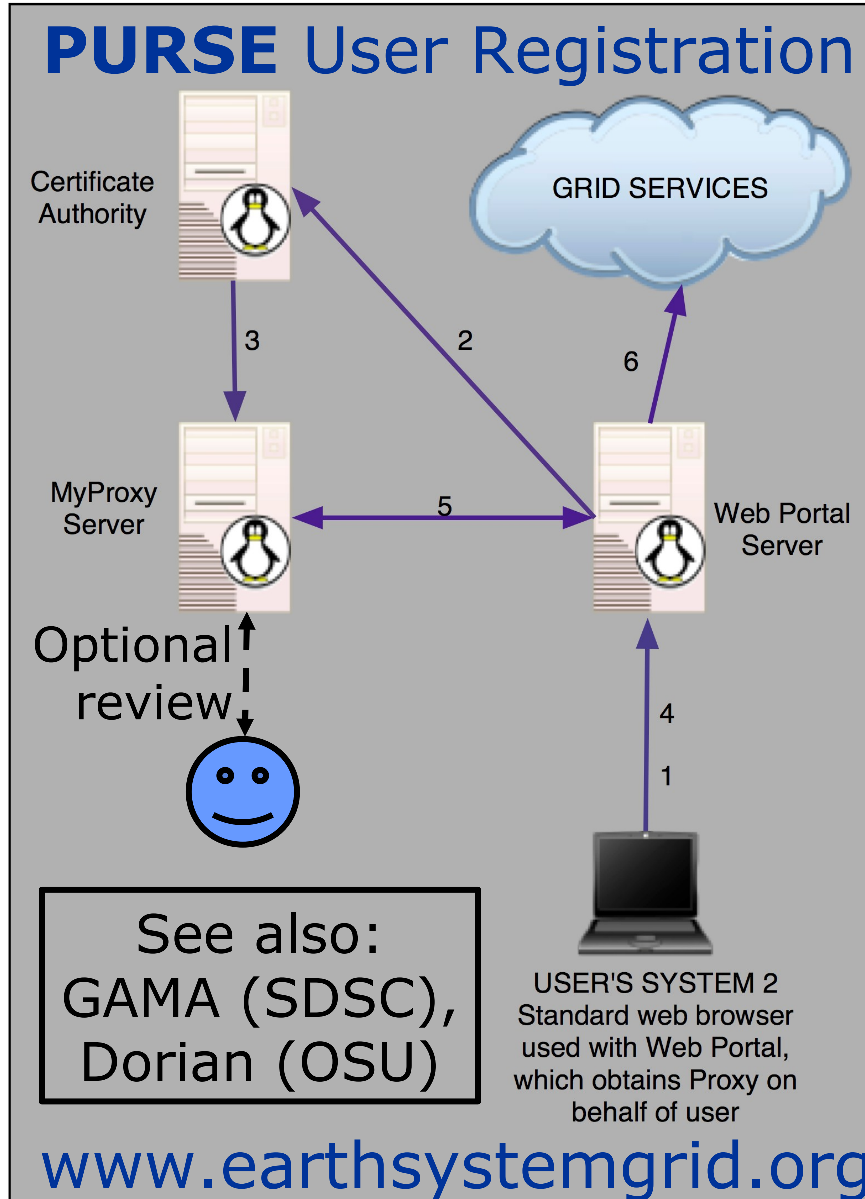
- Grid Services
  - GRAM
  - GridFTP (+striped GridFTP server)
  - MDS (+WebSDV, +Trigger Service, +Archiver)
  - MyProxy
  - SimpleCA
  - RLS
  - Catalog service
- Other Services
  - OpenDAPg
  - HPSS
  - SRM
  - Apache, Tomcat
- ESG-specific services
  - Workflow Manager
  - Registration Service

Under the Covers



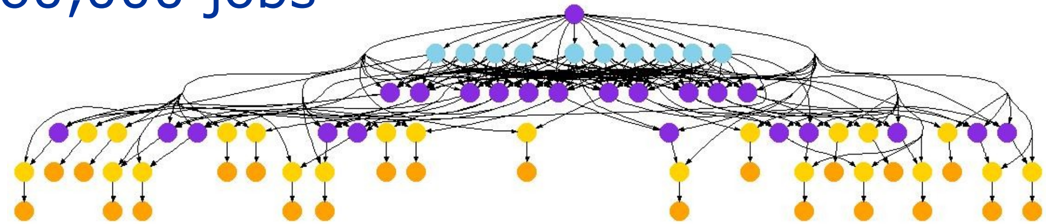
Security Needn't Be Hard: Earth System Grid

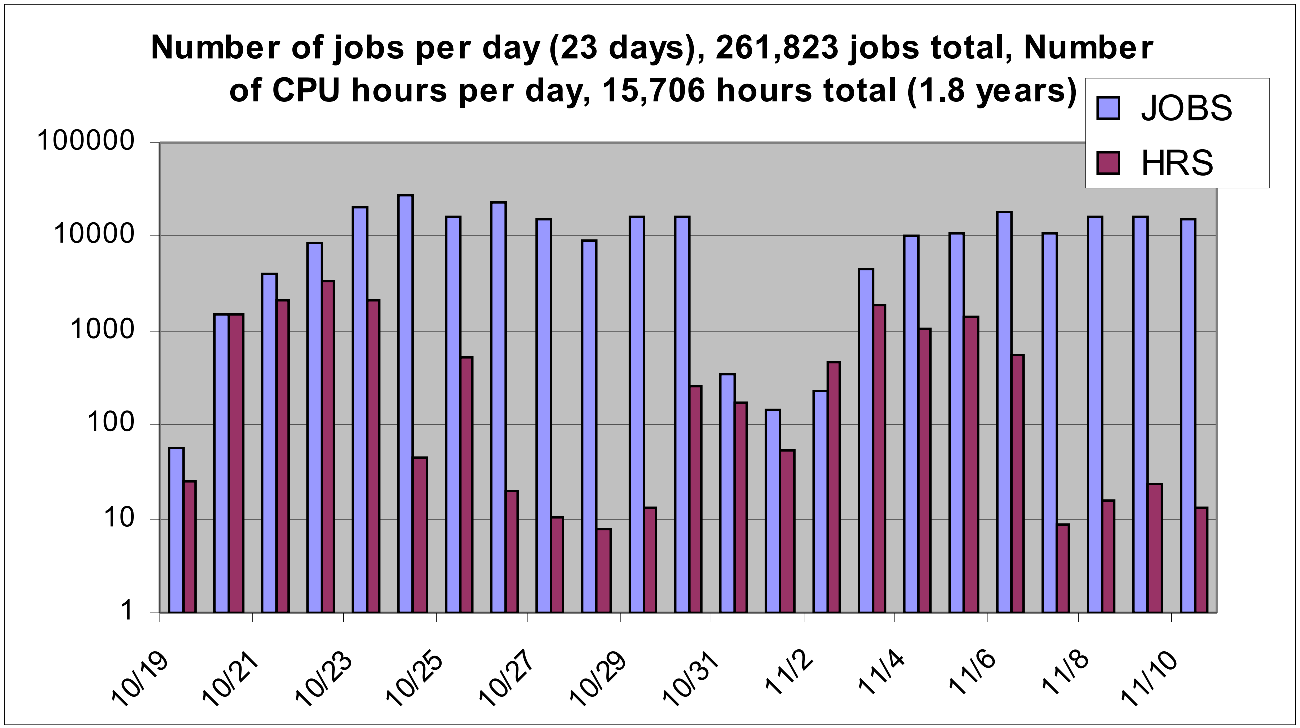
- Purpose
  - Access to large data
- Policies
  - Per-collection control
  - Different user classes
- Implementation (GT)
  - Portal-based User Registration Service
  - PKI, SAML assertions
- Experience
  - >2000 users
  - >100 TB downloaded



Southern California Earthquake Center (SCEC)

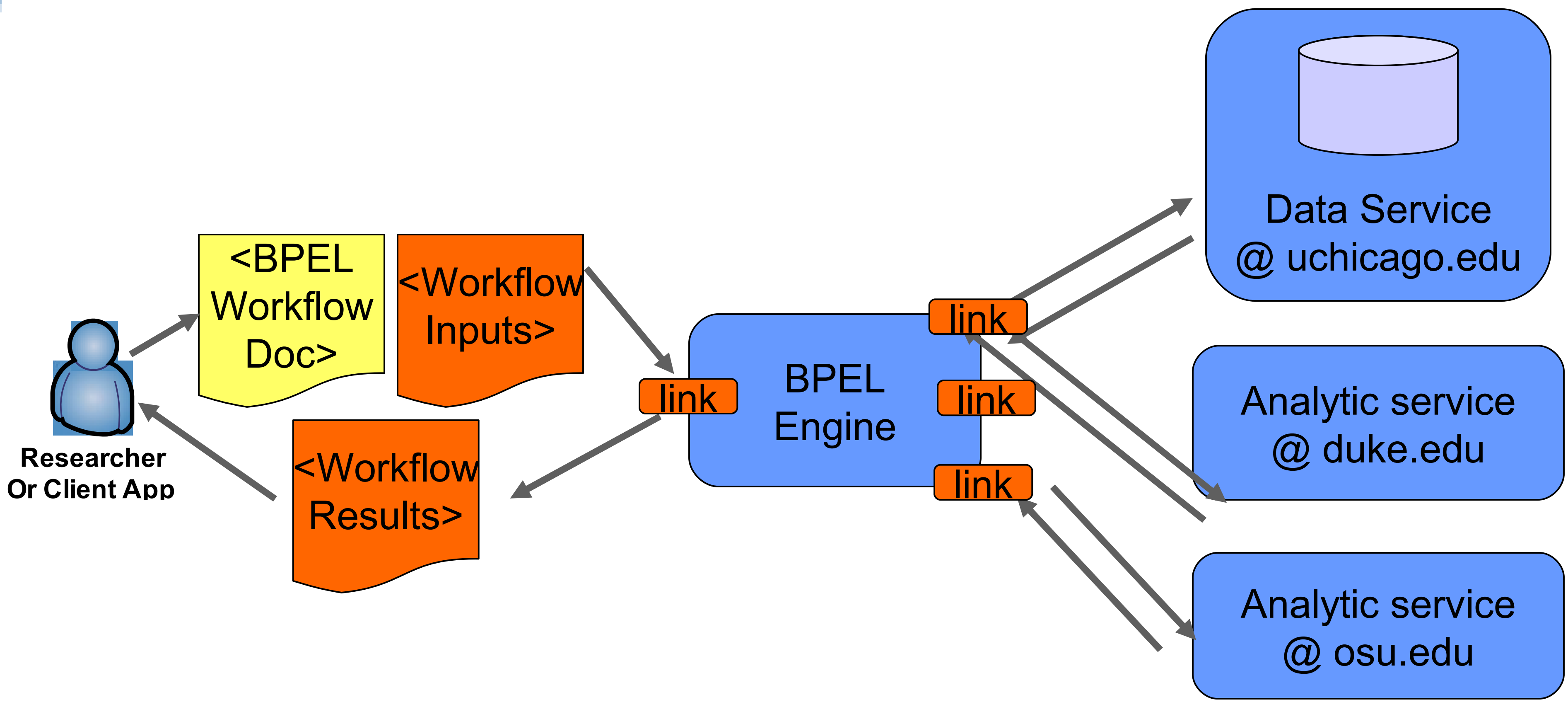
- Seismic hazard analysis application: used VDS services to manage 1.8 years of computation over 23 days to process 20 TB of data with 260,000 jobs





- Ewa Deelman et al., ISI
- In collaboration with Tom Jordan, Phil Maechlin, David Okaya (USC); Rob Graves (USGS) and others in SCEC

Example: Cancer Bioinformatics Grid



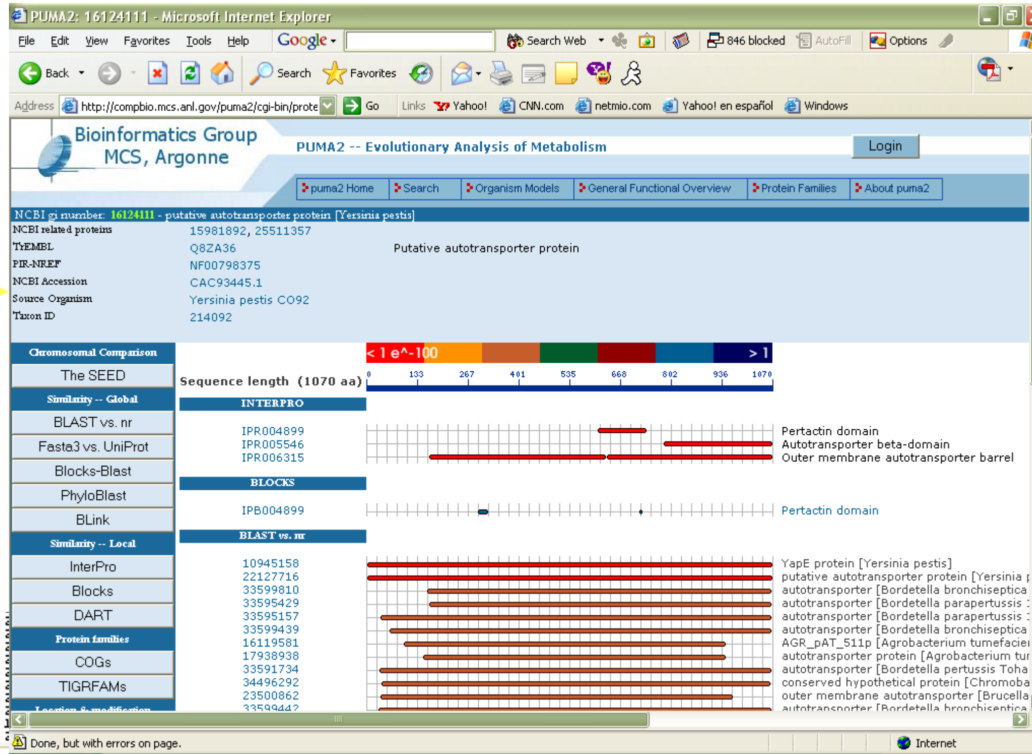
Each workflow is also a service, enacted by BPEL Engine

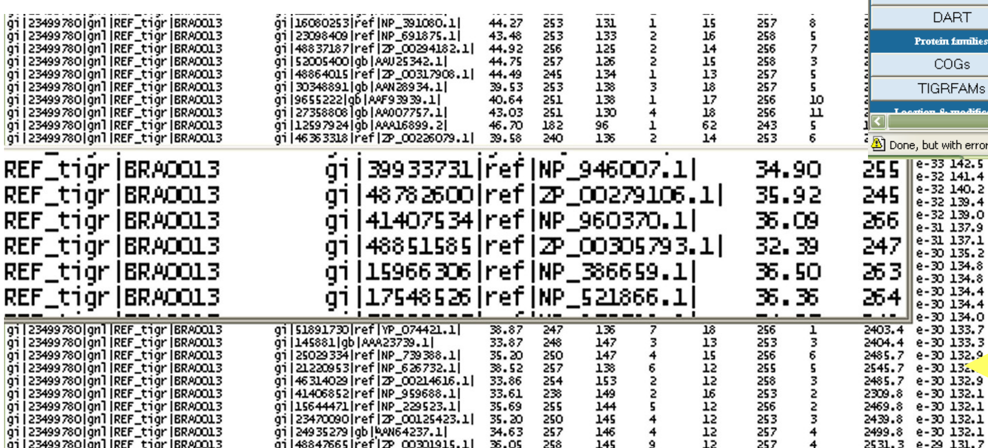
the globus alliance
www.globus.org

For Example: Biology

PUMA Knowledge Base

Information about proteins analyzed against ~2 million gene sequences





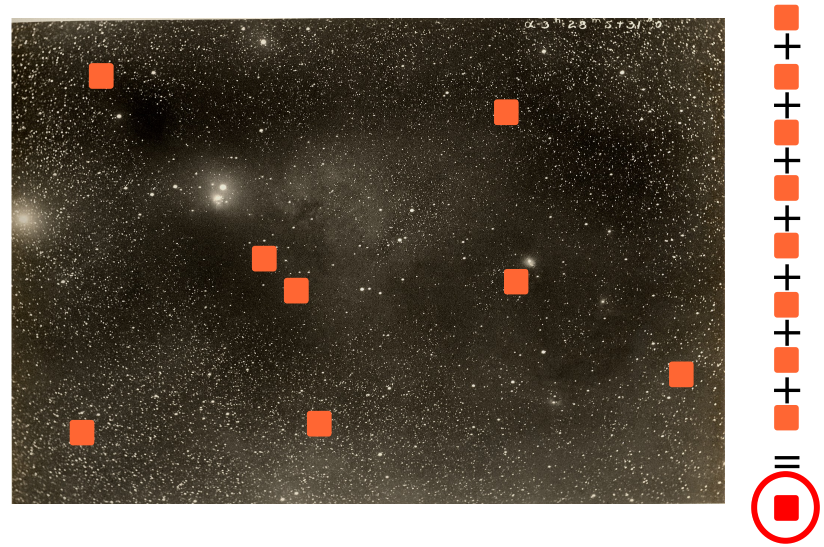
Analysis on Grid

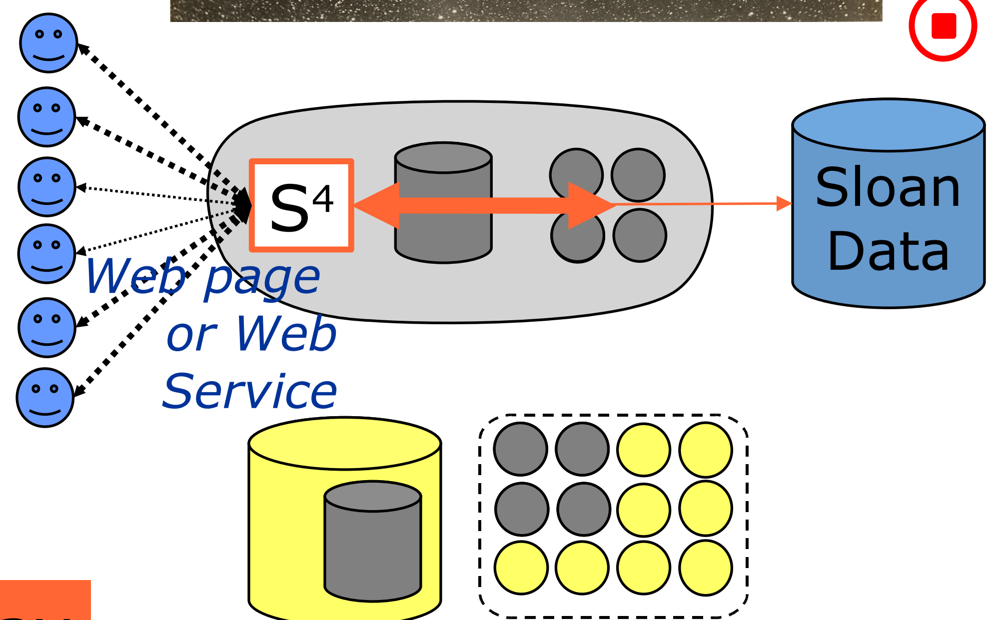
Involves millions of BLAST, BLOCKS, and other processes

Natalia Maltsev et al.
http://compbio.mcs.anl.gov/puma2

Astro Portal Stacking Service

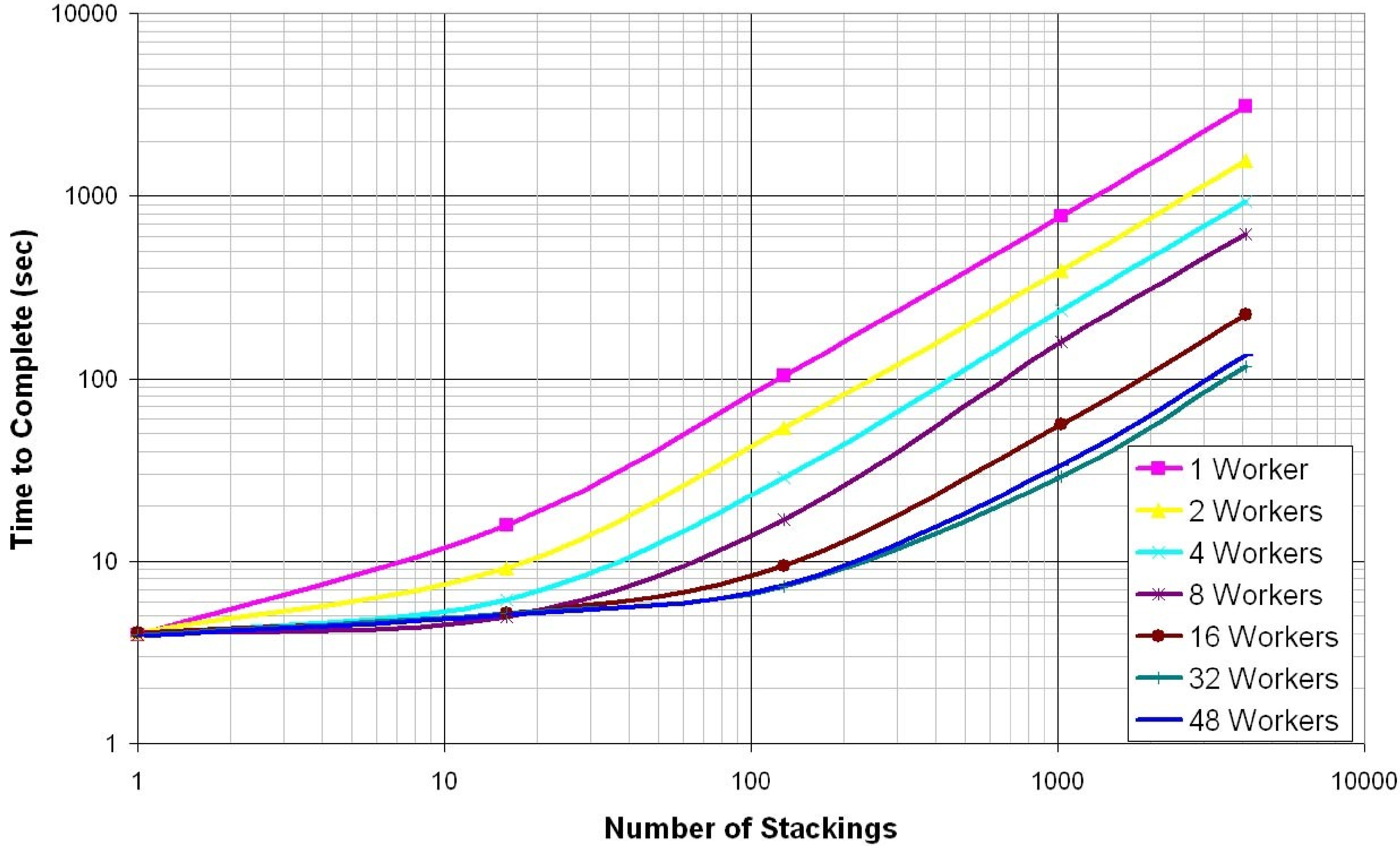
- Purpose
  - On-demand “stacks” of random locations within ~10TB dataset
- Challenge
  - Rapid access to 10-10K “random” files
  - Time-varying load
- Solution
  - Dynamic acquisition of compute, storage





With Ioan Raicu & Alex Szalay

Astro Portal Stacking Performance (LAN GPFS)



Summary

- Global data services
  - Connecting data with people & computers, often on a large scale
- Globus building blocks
  - Core Web Services & security; enabling mechanisms for data access & manipulation
- Building higher-level services
  - E.g., Data replication service
- Application case studies
- Summary

For More Information

- Globus Alliance
  - www.globus.org
- Dev.Globus
  - dev.globus.org
- Global Grid Forum
  - www.ggf.org
- TeraGrid
  - www.teragrid.org
- Open Science Grid
  - www.opensciencegrid.org
- Background information
  - www.mcs.anl.gov/~foster